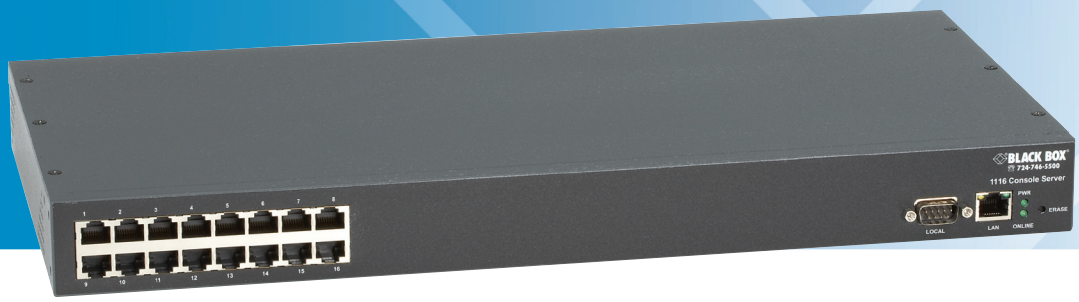
November 2009

LES1108A LES1208A
LES1116A LES1216A
LES1148A LES1248A

Value-Line and Advanced Console Servers User's Manual

Securely manage data center and network equipment from anywhere in the world.



Customer Support Information

Order toll-free in the U.S.: Call 877-877-BBOX (outside U.S. call 724-746-5500)
FREE technical support 24 hours a day, 7 days a week: Call 724-746-5500 or fax 724-746-0746
Mailing address: Black Box Corporation, 1000 Park Drive, Lawrence, PA 15055-1018
Web site: www.blackbox.com • E-mail: info@blackbox.com

Trademarks Used in this Manual

Black Box and the Double Diamond logo are registered trademarks of BB Technologies, Inc.

Mac is a registered trademark of Apple Computers, Inc.

Linux is a registered trademark of Linus Torvalds.

Internet Explorer, Windows, Windows Me, Windows NT, and Windows Vista are a registered trademarks of Microsoft Corporation.

Nagios is a registered trademark of Nagios Enterprises LLC.

Java and Solaris are trademarks of Sun Microsystems, Inc.

Unix is a registered trademark of X/Open Company Ltd.

Any other trademarks mentioned in this manual are acknowledged to be the property of the trademark owners.

We're here to help! If you have any questions about your application or our products, contact Black Box Tech Support at **724-746-5500** or go to **blackbox.com** and click on "Talk to Black Box." You'll be live with one of our technical experts in less than 20 seconds.

Federal Communications Commission and Industry Canada Radio Frequency Interference Statements

This equipment generates, uses, and can radiate radio-frequency energy, and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio communication. It has been tested and found to comply with the limits for a Class A computing device in accordance with the specifications in Subpart B of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when the equipment is operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever measures may be necessary to correct the interference.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This digital apparatus does not exceed the Class A limits for radio noise emission from digital apparatus set out in the Radio Interference Regulation of Industry Canada.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le Règlement sur le brouillage radioélectrique publié par Industrie Canada.

Instrucciones de Seguridad
(Normas Oficiales Mexicanas Electrical Safety Statement)

1. Todas las instrucciones de seguridad y operación deberán ser leídas antes de que el aparato eléctrico sea operado.
2. Las instrucciones de seguridad y operación deberán ser guardadas para referencia futura.
3. Todas las advertencias en el aparato eléctrico y en sus instrucciones de operación deben ser respetadas.
4. Todas las instrucciones de operación y uso deben ser seguidas.
5. El aparato eléctrico no deberá ser usado cerca del agua—por ejemplo, cerca de la tina de baño, lavabo, sótano mojado o cerca de una alberca, etc..
6. El aparato eléctrico debe ser usado únicamente con carritos o pedestales que sean recomendados por el fabricante.
7. El aparato eléctrico debe ser montado a la pared o al techo sólo como sea recomendado por el fabricante.
8. Servicio—El usuario no debe intentar dar servicio al equipo eléctrico más allá a lo descrito en las instrucciones de operación. Todo otro servicio deberá ser referido a personal de servicio calificado.
9. El aparato eléctrico debe ser situado de tal manera que su posición no interfiera su uso. La colocación del aparato eléctrico sobre una cama, sofá, alfombra o superficie similar puede bloquea la ventilación, no se debe colocar en libreros o gabinetes que impidan el flujo de aire por los orificios de ventilación.
10. El equipo eléctrico deber ser situado fuera del alcance de fuentes de calor como radiadores, registros de calor, estufas u otros aparatos (incluyendo amplificadores) que producen calor.
11. El aparato eléctrico deberá ser connectado a una fuente de poder sólo del tipo descrito en el instructivo de operación, o como se indique en el aparato.
12. Precaución debe ser tomada de tal manera que la tierra fisica y la polarización del equipo no sea eliminada.
13. Los cables de la fuente de poder deben ser guiados de tal manera que no sean pisados ni pellizcados por objetos colocados sobre o contra ellos, poniendo particular atención a los contactos y receptáculos donde salen del aparato.
14. El equipo eléctrico debe ser limpiado únicamente de acuerdo a las recomendaciones del fabricante.
15. En caso de existir, una antena externa deberá ser localizada lejos de las lineas de energia.
16. El cable de corriente deberá ser desconectado del cuando el equipo no sea usado por un largo periodo de tiempo.
17. Cuidado debe ser tomado de tal manera que objectos liquidos no sean derramados sobre la cubierta u orificios de ventilación.
18. Servicio por personal calificado deberá ser provisto cuando:
    - A: El cable de poder o el contacto ha sido dañado; u
    - B: Objectos han caído o líquido ha sido derramado dentro del aparato; o
    - C: El aparato ha sido expuesto a la lluvia; o
    - D: El aparato parece no operar normalmente o muestra un cambio en su desempeño; o
    - E: El aparato ha sido tirado o su cubierta ha sido dañada.

INDEX

INTRODUCTION 11
INSTALLATION 15
2.1 Models 15
2.1.1 Kit components LES1208A, LES1216A and LES1248A Advanced Console Servers *16*
2.1.2 Kit components LES1116A and LES1148A Console Servers *16*
2.1.3 Kit components LES1108A Console Server *17*
2.2 Power connection 17
2.2.1 LES1208A, LES1216A and LES1248A power *17*
2.2.2 LES1116A and LES1148A power *17*
2.2.3 LES1108A power *18*
2.3 Network connection 18
2.4 Serial Port connection 18
2.5 USB Port Connection 19
SYSTEM CONFIGURATION 20
3.1 Management console connection 20
3.1.1 Connected PC/workstation set up *20*
3.1.2 Browser connection *21*
3.2 *Administrator* Password 22
3.3 Network IP address 23
3.3.1 IPv6 configuration *25*
3.4 System Services 25
3.5 Communications Software 27
3.5.1 SDT Connector *27*
3.5.2 PuTTY *28*
3.5.3 SSHTerm *28*
3.6 Management network configuration (LES1208A, LES1216A and LES1248A only) 29
3.6.1 Enable the Management LAN *29*
3.6.2 Configure the DHCP server *30*
3.6.3 Select Failover or broadband OOB *32*
3.6.4 Bridging the network ports *33*
SERIAL PORT AND NETWORK HOST 35
4.1 Configure Serial Ports 35
4.1.1 Common Settings *36*
4.1.2 Console Server Mode *37*
4.1.3 SDT Mode *42*
4.1.4 Device (RPC, UPS, EMD) Mode *43*
4.1.5 Terminal Server Mode *43*
4.1.6 Serial Bridging Mode *43*
4.1.8 Syslog *44*
4.2 Add/ Edit Users 45
4.3 Authentication 47
4.4 Network Hosts 47
4.5 Trusted Networks 48
4.6 Serial Port Cascading 50
4.6.1 Automatically generate and upload SSH keys *50*
4.6.2 Manually generate and upload SSH keys *51*
4.6.3 Configure the slaves and their serial ports *53*
4.6.4 Managing the Slaves *54*
4.7 Serial Port Redirection 54
4.8 Managed Devices 55

FAILOVER AND OoB DIAL-IN 58
5.1 OoB Dial-In Access 58
5.1.1 Configure Dial-In PPP *59*
5.1.2 Using SDT Connector client *61*
5.1.3 Set up Windows XP/ 2003/Vista/7 client *61*
5.1.4 Set up earlier Windows clients *62*
5.1.5 Set up Linux clients for dial-in *62*
5.2 OoB broadband access 62
5.3 Broadband Ethernet Failover 62
5.4 Dial-Out Failover 63
SECURE SSH TUNNELING AND SDT CONNECTOR 65
6.1 Configuring for SSH Tunneling to Hosts 66
6.2 SDT Connector Client Configuration 66
6.2.1 SDT Connector installation *67*
6.2.2 Configuring a new console server gateway in the SDT Connector client *68*
6.2.3 Auto-configure SDT Connector client with the user's access privileges *69*
6.2.4 Make an SDT connection through the gateway to a host *70*
6.2.5 Manually adding hosts to the SDT Connector gateway *71*
6.2.6 Manually adding new services to the new hosts *72*
6.2.7 Adding a client program to be started for the new service *74*
6.2.8 Dial in configuration *76*
6.3 SDT Connector to Management Console 76
6.4 SDT Connector - telnet or SSH connect to serially attached devices 77
6.5 Using SDT Connector for out-of-band connection to the gateway 79
6.6 Importing (and exporting) preferences 80
6.7 SDT Connector Public Key Authentication 81
6.8 Setting up SDT for Remote Desktop access 81
6.8.1 Enable Remote Desktop on the target Windows computer to be accessed *82*
6.8.2 Configure the Remote Desktop Connection client *83*
6.9 SDT SSH Tunnel for VNC 87
6.9.1 Install and configure the VNC Server on the computer to be accessed *87*
6.9.2 Install, configure and connect the VNC Viewer *88*
6.10 Using SDT to IP connect to hosts that are serially attached to the gateway 90
6.10.1 Establish a PPP connection between the host COM port and console server *90*
6.10.2 Set up SDT Serial Ports on console server *93*
6.10.3 Set up SDT Connector to SSH port forward over the console server Serial Port *94*
6.11 SSH Tunneling using other SSH clients (e.g. PuTTY) 94
ALERTS AND LOGGING 98
7.1 Configure SMTP/SMS/SNMP/Nagios alert service 98
7.1.1 Email alerts *98*
7.1.2 SMS alerts *99*
7.1.3 SNMP alerts *100*
7.1.4 Nagios alerts *101*
7.2 Activate Alert Events and Notifications 101
7.2.1 Add a new alert *102*
7.2.2 Configuring general alert types *103*
7.2.3 Configuring environment and power alert type *104*
7.2.4 Configuring alarm sensor alert type *105*
7.3 Remote Log Storage 105
7.4 Serial Port Logging 106
7.5 Network TCP or UDP Port Logging 106
POWER & ENVIRONMENTAL MANAGEMENT 108
8.1 Remote Power Control (RPC) 108

8.1.1 RPC connection 108
8.1.2 RPC access privileges and alerts 111
8.1.3 User power management 111
8.1.4 RPC status 112
8.2 Uninterruptible Power Supply Control (UPS) 112
8.2.1 Managed UPS connections 113
8.2.2 Remote UPS management 116
8.2.3 Controlling UPS powered computers 117
8.2.4 UPS alerts 118
8.2.5 UPS status 118
8.2.6 Overview of Network UPS Tools (NUT) 119
8.3 Environmental Monitoring 121
8.3.1 Connecting the EMD 122
8.3.2 Environmental alerts 123
8.3.3 Environmental status 123
AUTHENTICATION 125
9.1 Authentication Configuration 125
9.1.1 Local authentication 126
9.1.2 TACACS authentication 126
9.1.3 RADIUS authentication 127
9.1.4 LDAP authentication 128
9.1.5 RADIUS/TACACS user configuration 128
9.2 PAM (Pluggable Authentication Modules) 129
9.3 SSL Certificate 131
NAGIOS INTEGRATION 134
10.1 Nagios Overview 134
10.2 Central management and setting up SDT for Nagios 135
10.2.1 Set up central Nagios server 136
10.2.2 Set up distributed console servers 136
10.3 Configuring Nagios distributed monitoring 139
10.3.1 Enable Nagios on the console server 139
10.3.2 Enable NRPE monitoring 140
10.3.3 Enable NSCA monitoring 140
10.3.4 Configure selected Serial Ports for Nagios monitoring 141
10.3.5 Configure selected Network Hosts for Nagios monitoring 141
10.3.6 Configure the upstream Nagios monitoring host 142
10.4 Advanced Distributed Monitoring Configuration 142
10.4.1 Sample Nagios configuration 142
10.4.2 Basic Nagios plug-ins 145
10.4.3 Additional plug-ins 146
10.4.4 Number of supported devices 146
10.4.5 Distributed Monitoring Usage Scenarios 147
SYSTEM MANAGEMENT 150
11.1 System Administration and Reset 150
11.2 Upgrade Firmware 151
11.3 Configure Date and Time 152
11.4 Configuration Backup 153
STATUS REPORTS 156
12.1 Port Access and Active Users 156
12.2 Statistics 157
12.3 Support Reports 157
12.4 Syslog 158
12.5 Dashboard 158

12.5.1 Configuring the Dashboard 159
12.5.2 Creating custom widgets for the Dashboard 161
MANAGEMENT 162
13.1 Device Management 162
13.2 Port and Host Logs 163
13.3 Serial Port Terminal Connection 163
13.4 Power Management 164
CONFIGURATION FROM THE COMMAND LINE 165
14.1 Accessing *config* from the command line 165
14.2 Serial Port configuration 168
14.3 Adding and removing Users 171
14.4 Adding and removing user Groups 172
14.5 Authentication 173
14.6 Network Hosts 174
14.7 Trusted Networks 175
14.8 Cascaded Ports 176
14.9 UPS Connections 176
14.10 RPC Connections 177
14.11 Environmental 178
14.12 Managed Devices 179
14.13 Port Log 179
14.14 Alerts 180
14.15 SMTP & SMS 182
14.16 SNMP 183
14.17 Administration 183
14.18 IP settings 184
14.19 Date & Time settings 184
14.20 Dial-in settings 185
14.21 DHCP server 186
14.22 Services 186
14.23 NAGIOS 187
ADVANCED CONFIGURATION 189
15.1 Custom Scripting 189
15.1.1 Custom script to run when booting 189
15.1.2 Running custom scripts when alerts are triggered 190
15.1.3 Example script - Power cycling on pattern match 191
15.1.4 Example script - Multiple email notifications on each alert 191
15.1.5 Deleting configuration values from the CLI 191
15.1.6 Power cycle any device upon a ping request failure 194
15.1.7 Running custom scripts when a configurator is invoked 196
15.1.8 Backing-up the configuration and restoring using a local USB stick 196
15.1.9 Backing-up the configuration off-box 197
15.2 Advanced Portmanager 198
15.2.1 Portmanager commands 198
15.2.2 External Scripts and Alerts 199
15.3 Raw Access to Serial Ports 200
15.3.1 Access to serial ports 200
15.3.2 Accessing the console/modem port 201
15.4 IP- Filtering 201
15.5 Modifying SNMP Configuration 202
15.5.1 /etc/config/snmpd.conf 202
15.5.2 Adding more than one SNMP server 203
15.6 Secure Shell (SSH) Public Key Authentication 204

15.6.1 SSH Overview *204*
15.6.2 Generating Public Keys (Linux) *205*
15.6.3 Installing the SSH Public/Private Keys (Clustering) *205*
15.6.4 Installing SSH Public Key Authentication (Linux) *206*
15.6.5 Generating public/private keys for SSH (Windows) *207*
15.6.6 Fingerprinting *209*
15.6.7 SSH tunneled serial bridging *210*
15.6.8 SDT Connector Public Key Authentication *212*
15.7 Secure Sockets Layer (SSL) Support 213
15.8 HTTPS 213
15.8.1 Generating an encryption key *213*
15.8.2 Generating a self-signed certificate with OpenSSL *213*
15.8.3 Installing the key and certificate *214*
15.8.4 Launching the HTTPS Server *214*
15.9 Power Strip Control 215
15.9.1 The PowerMan tool *215*
15.9.2 The pmpower tool *216*
15.9.3 Adding new RPC devices *217*
15.10 IPMItool 218
15.11 Custom Development Kit (CDK) 221
15.12 Scripts for Managing Slaves 222

APPENDIX

A. CLI Commands and Source Code

B. Hardware Specification

C. Safety and Certifications

D. Connectivity and Serial I/O

E. Terminology

F. End User License Agreement

G. Service and Warranty

Chapter 1 Introduction

This Manual

This User's Manual walks you through installing and configuring your Black Box Console Server (LES1108A, LES1116A, LES1148A) or Advanced Console Server (LES1208A, LES1216A, LES1248A). Each of these products is referred to generically in this manual as a "*console server."*

Once configured, you will be able to use your *console server* to securely monitor access and control the computers, networking devices, telecommunications equipment, power-supplies, and operating environments in your data room or communications centers. This manual guides you in managing this infrastructure locally (across your operations or management LAN or through the local serial console port), and remotely (across the Internet, private network, or via dial up).

Manual Organization

This manual contains the following chapters:

1. Introduction	An overview of the features of *console server* and information on this manual.
2. Installation	Physical installation of the *console server* and how to interconnect controlled devices.
3. System Configuration	Describes the initial installation and configuration using the Management Console. Covers configuration of the *console server* on the network and the services that will be supported.
4. Serial & Network	Covers configuring serial ports and connected network hosts, and setting up Users and Groups.
5. Failover and OoB dial-in	Describes setting up the high availability access features of the *console server*.
6. Secure Tunneling (SDT)	Covers secure remote access using SSH and configuring for RDP, VNC, HTTP, HTTPS, etc. access to network and serially connected devices.
7. Alerts and Logging	Explains how to set up local and remote event/data logs and how to trigger SNMP and email alerts.
8. Power & Environment	Describes how to manage USB, serial, and network attached power strips and UPS supplies including Network UPS Tool (NUT) operation, IPMI power control, and EMD environmental sensor configuration.
9. Authentication	Access to the *console server* requires usernames and passwords that are locally or externally authenticated.
10. Nagios Integration	Describes how to set Nagios central management with SDT extensions and configure the *console server* as a distributed Nagios server.
11. System Management	Covers access to and configuration of services that will run on the *console server.*

12. Status Reports	View a dashboard summary and detailed status and logs of serial and network connected devices (ports, hosts, power, and environment)
13. Management	Includes port controls that *Users* can access.
14 Basic Configuration	Command line installation and configuration using the *config* command.
15. Advanced Config	More advanced command line configuration activities where you will need to use Linux commands.

The latest update of this manual can be found online at *www.*Black Box*.com/download.html*

Types of users

The *console server* supports two classes of users:

I. First, there are the administrative users who will be authorized to configure and control the *console server*; and to access and control all the connected devices. These administrative users will be set up as members of the **admin** user group and any user in this class is referred to generically in this manual as the ***Administrator***. An *Administrator* can access and control the *console server* using the *config* utility, the Linux command line, or the browser-based Management Console. By default, the *Administrator* has access to all services and ports to control all the serial connected devices and network connected devices (*hosts*).

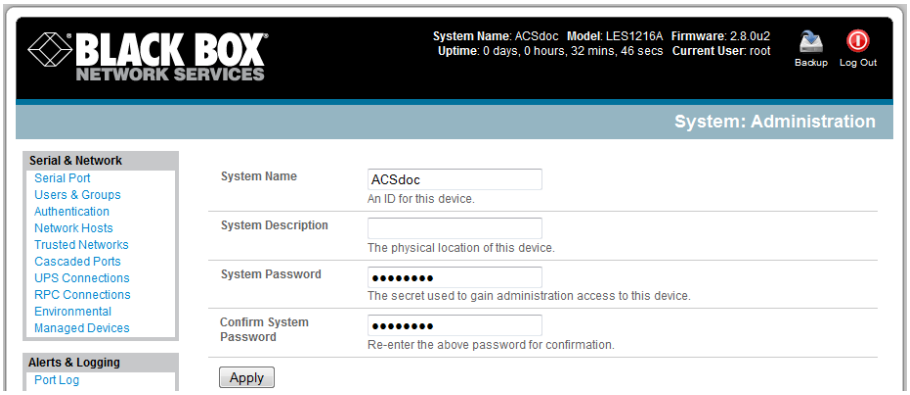
II. The second class of users are those who have been set up by the *Administrator* with specific limits of their access and control authority. These users are set up as members of the **users** user group (or some other user groups the *Administrator* may have added). They are only authorized to perform specified controls on specific connected devices and are referred to as ***Users***. These *Users* (when authorized) can access serial or network connected devices; and control these devices using the specified services (for example, Telnet, HHTPS, RDP, IPMI, Serial over LAN, Power Control). An authorized *User* also has a limited view of the Management Console and can only access authorized configured devices and review port logs.

In this manual, when the term ***user*** (lower case) is used, it refers to both the above classes of users. This document also uses the term ***remote users*** to describe users who are not on the same LAN segment as the *console server*. These remote users may be *Users,* who are on the road connecting to managed devices over the public Internet, or it may be an *Administrator* in another office connecting to the *console server* itself over the enterprise VPN, or the remote user may be in the same room or the same office but connected on a separate VLAN than the *console server*.

Management Console

The Management Console provides a view of the *console server* and all the connected devices.

Administrators can use any browser to log into the Management Console either locally or from a remote location. They can then use Management Console to manage the *console server*, the users, the serial ports and serially connected devices, network connected hosts, and connected power devices; and to view associated logs and configure alerts.



A *User* can also use the Management Console, but has limited menu access to control select devices, review their logs and access them using the built-in java terminal or control power to them.

The *console server* runs an embedded Linux operating system, and experienced Linux® and UNIX® users may prefer to configure it at the command line. To get command line access, connect through a terminal emulator or communications program to the console serial port; connect via ssh or telnet through the LAN; or connect through an SSH tunneling to the *console server*.

Manual Conventions

This manual uses different fonts and typefaces to show specific actions:

Note Text presented like this indicates issues to note.

Text presented like this highlights important information. Make sure you read and follow these warnings.

- Text presented with an arrow head indent indicates an action you should take as part of the procedure.

Bold text indicates text that you type, or the name of a screen object (*for example,* a menu or button) on the Management Console.

Italic text indicates a text command you enter at the command line level.

Publishing history

Date	Revision	Update details
September 2009	0.9	Prelease

Copyright

©Black Box Corporation 2009. All Rights Reserved.

Information in this document is subject to change without notice and does not represent a commitment on the part of Black Box. Black Box provides this document “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose.

Black Box may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time. This manual could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Notice to Users

Use proper back-up systems and necessary safety devices to protect against injury, death, or property damage caused by system failure. This protection is the user’s responsibility.

This device is not approved for use as a life-support or medical system.

Any changes or modifications made to this device without the explicit approval or consent of Black Box will void Black Box of any liability or responsibility of injury or loss caused by any malfunction.

This equipment is for indoor use and all the communication wirings are limited to the inside of the building.

Chapter 2 Installation

Introduction

This chapter describes how to install the *console server* hardware and connect it to controlled devices.

To avoid physical and electrical hazards please read Appendix C on Safety.

2.1 Models

There are multiple *console server* models, each with a different number of network and serial ports or power supply configurations:

	Serial Ports	USB Ports	Network Ports	Console Port	Modem	RJ Pinout	Power	Memory (flash/RAM)
LES1248A	48	1	2	1	Internal	01	Dual DC	16/64MB
LES1216A	16	1	2	1	Internal	01	Dual AC	16/64MB
LES1208A	8	1	2	1	Internal	01	Dual AC	16/64MB
LES1148A	48	-	1	1	-	00	Single AC	16/64MB
LES1116A	16	-	1	1	-	00	Single AC	16/64MB
LES1108A	8	-	1	1	-	00	Ext AC/DC	8/16MB

The next sections show the components shipped with each of these models.

- Unpack your kit and verify you have all the parts shown above, and that they all appear in good working order.
- If you are installing the console server in a rack, you will need to attach the rack mounting brackets supplied with the unit, then install the unit in the rack. Make sure you follow the Safety Precautions listed in Appendix C.
- Connect your *console server* to the network, to the serial ports of the controlled devices, and to power as outlined next.

2.1.1 Kit components LES1208A, LES1216A and LES1248A Advanced Console Servers

LES1208A, LES1216A, or LES1248A Advanced Console Server

(2) UTP CAT5 blue cables

DB9F-RJ45S straight and DB9F-RJ45S cross-over connectors

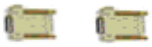
Dual IEC AC power cords

Printed Quick Start Guide and User's Manual on CD-ROM

2.1.2 Kit components LES1116A and LES1148A Console Servers

LES1116A or LES1148A Console Server

(2) UTP CAT5 blue cables



DB9F-RJ45S straight and DB9F-RJ45S cross-over connectors

IEC AC power cord

Printed Quick Start Guide and User's Manual on CD-ROM

2.1.3 Kit components LES1108A Console Server

LES1108A Console Server

(2) UTP CAT5 blue cables

DB9F-RJ45S straight and DB9F-RJ45S cross-over connectors

5-VDC, 2.0A, Power Supply with IEC Socket and AC power cable

Printed Quick Start Guide and this User's Manual on CD-ROM

2.2 Power connection

2.2.1 LES1208A, LES1216A and LES1248A power

The LES1208A, LES1216A and LES1248A *console server*s all have dual universal AC power supplies with auto failover built in. These power supplies each accept AC input voltage between 100 and 240 VAC with a frequency of 50 or 60 Hz. The total power consumption per *console server* is less than 30W. Two IEC AC power sockets are located at the rear of the metal case, and these IEC power inlets use conventional IEC AC power cords. Power cords for various regions are available, although the North American power cord is provided by default. There is a warning notice printed on the back of each unit.

To avoid electrical shock, connect the power cord grounding conductor to ground!

2.2.2 LES1116A and LES1148A power

The LES1116A and LES1148A models have a built-in universal auto-switching AC power supply. This power supply accepts AC input voltage between 100 and 240 VAC with a frequency of 50 or 60 Hz. The power consumption is less than 20W.

Both LES1116A and LES1148A models have an IEC AC power socket located in the rear of the metal case. This IEC power inlet uses a conventional IEC AC power cord, and the power cords for various regions are available. Call Black Box Technical Support for details at 724-746-5500. (The North American power cord is provided by default.) There is a warning notice printed on the back of each unit.

To avoid electrical shock, connect the power cord grounding conductor to ground.

2.2.3 LES1108A power

The LES1108A includes an external DC power supply unit. This unit accepts an AC input voltage between 100 and 250 VAC with a frequency of 50Hz or 60Hz. The DC power supply has an IEC AC power socket, which accepts a conventional IEC AC power cord. The power cord for North America is included in the kit. The 5-VDC connector from the power supply plugs into the 5VDC power socket on the rear of the LES1108A.

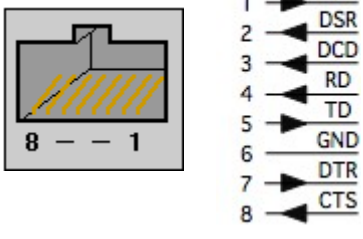
2.3 Network connection

The RJ-45 LAN ports are located on the rear panel of the LES1108A and on the front panel of the rack-mount *console server*s. Use industry standard Cat5 cabling and connectors. Make sure that you only connect the LAN port to an Ethernet network that supports 10BASE-T/100BASE-T. To initially configure the console server, you must connect a PC or workstation to the *console server*'s principal network port (labeled *NETWORK1* or *LAN*).

2.4 Serial Port connection

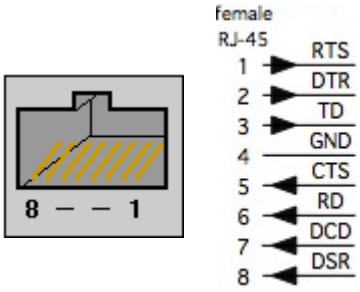
The RJ-45 serial ports are located on the LES1108A's rear panel and on the rackmount *console servers*' front panel.

The LES1108A, LES1116A and LES1148A Console Servers have the Black Box Classic RJ-45 pinout shown below:



PIN	SIGNAL	DEFINITION	DIRECTION
1	RTS	Request To Send	Output
2	DSR	Data Set Ready	Input
3	DCD	Data Carrier Detect	Input
4	RXD	Receive Data	Input
5	TXD	Transmit Data	Output
6	GND	Signal Ground	NA
7	DTR	Data Terminal Ready	Output
8	CTS	Clear To Send	Input

The LES1208A, LES1216A, and LES1248A Advanced Console Servers have the Cyclades RJ-45 pinout shown next:



PIN	SIGNAL	DEFINITION	DIRECTION
1	RTS	Request To Send	Output
2	DTR	Data Terminal Ready	Output
3	TXD	Transmit Data	Output
4	GND	Signal Ground	NA
5	CTS	Clear To Send	Input
6	RXD	Receive Data	Input
7	DCD	Data Carrier Detect	Input
8	DSR	Data Set Ready	Input

The *console servers* also have a DB9 LOCAL (Console/Modem) port that is on the LE1108A's rear panel and on the rackmount units' front panels.

Conventional CAT5 cabling with RJ-45 jacks is used for serial connections. Before connecting an external device's console port to the *console server* serial port, confirm that the device supports the standard RS-232C (EIA-232).

Black Box supplies a range of cables and adapters that may be required to connect to the more popular servers and network appliances. Call Technical Support at 724-746-5500 for details.

2.5 USB Port Connection

The LES1208A, LES1216A and LES1248A *console server*s each also have one USB port. These *console server*s ship with a USB memory. Install the memory stick in the USB port to store log files.

Chapter 3 Initial System Configuration

Introduction

This chapter provides step-by-step instructions for the console server's initial configuration, and for connecting it to the Management or Operational LAN. The *Administrator* must:

- Activate the Management Console.
- Change the *Administrator* password.
- Set the IP address *console server*'s principal LAN port.
- Select the network services that will be supported.

This chapter also discusses the communications software tools that the *Administrator* may use to access the *console server*.

3.1 Management console connection

Your *console server* is configured with a default IP Address 192.168.0.1 Subnet Mask 255.255.255.0

- Directly connect a PC or workstation to the *console server*.

Note For initial configuration we recommend that you connect the *console server* directly to a single PC or workstation. However, if you choose to connect your LAN before completing the initial setup steps, it is important that:

- you make sure that there are no other devices on the LAN with an **address of 192.168.0.1**
- the *console server* and the PC/workstation are on the same LAN segment, with no interposed router appliances.

3.1.1 Connected PC/workstation set up

To configure the *console server* with a browser, the connected PC/workstation should have an IP address in the same range as the *console server* (e.g. 192.168.0.100):

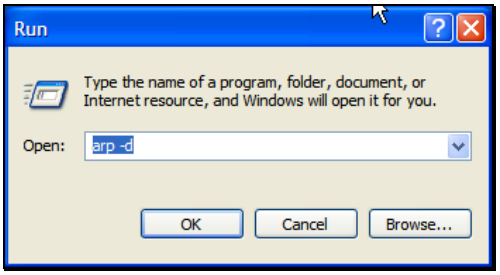
- To configure the IP Address of your Linux or Unix PC/workstation simply run *ifconfig*
- For Windows PCs (Win9x/Me/2000/XP/ Vista/ NT):
  - Click **Start** -> (**Settings** ->) **Control Panel** and double click **Network Connections** (for 95/98/Me, double click **Network**).
  - Right click on **Local Area Connection** and select **Properties.**
  - Select **Internet Protocol (TCP/IP)** and click **Properties.**
  - Select **Use the following IP address** and enter the following details:
    - IP address: **192.168.0.100**

- Subnet mask: **255.255.255.0**

- If you want to retain your existing IP settings for this network connection, click **Advanced** and **Add** the above as a secondary IP connection.

➢ If it is not convenient to change your PC/workstation network address, you can use the *ARP-Ping* command to reset the *console server* IP address. To do this from a Windows PC:

- Click **S**tart -> R**un** (or select **All Programs** then **Accessories** then **Run**).
- Type *cmd* and click **OK** to bring up the command line.
- Type *arp –d* to flush the ARP cache.
- Type *arp –a* to view the current ARP cache (this should be empty).



Now add a static entry to the ARP table and *ping* the *console server* to assign the IP address to the console server. In the example below, a *console server* has a MAC Address 00:13:C6:00:02:0F (designated on the label on the bottom of the unit) and we are setting its IP address to 192.168.100.23. Also the PC/workstation issuing the *arp* command must be on the same network segment as the *console server* (that is, have an IP address of 192.168.100.xxx)

- Type *arp -s 192.168.100.23 00-13-C6-00-02-0F* (Note for UNIX the syntax is: *arp -s 192.168.100.23 00:13:C6:00:02:0F*).
- Type *ping -t 192.18.100.23* to start a continuous ping to the new IP Address.
- Turn on the *console server* and wait for it to configure itself with the new IP address. It will start replying to the ping at this point.
- Type *arp –d* to flush the ARP cache again.

3.1.2 Browser connection

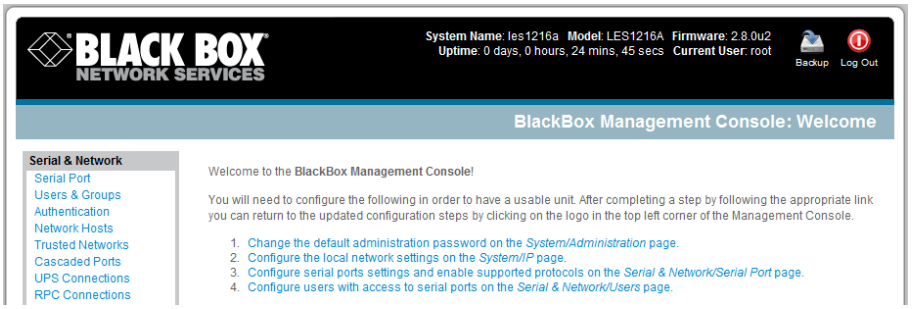
➢ Activate your preferred browser on the connected PC/workstation and enter **https://192.168.0.1** The Management Console supports all current versions of the popular browsers (Internet Explorer, Mozilla Firefox, Chrome, and more).

- You will be prompted to log in. Enter the default administration username and administration password:

  Username: **root**

  Password: **default**

Note *Console servers* are factory configured with HTTPS access enabled and HTTP access disabled.


BLACK BOX
NETWORK SERVICES
System Name: les1216a Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 0 hours, 24 mins, 45 secs Current User: root
Backup
Log Out
BlackBox Management Console: Welcome
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Welcome to the BlackBox Management Console!
You will need to configure the following in order to have a usable unit. After completing a step by following the appropriate link you can return to the updated configuration steps by clicking on the logo in the top left corner of the Management Console.
1. Change the default administration password on the System/Administration page.
2. Configure the local network settings on the System/IP page.
3. Configure serial ports settings and enable supported protocols on the Serial & Network/Serial Port page.
4. Configure users with access to serial ports on the Serial & Network/Users page.

A **Welcome** screen, which lists four initial installation configuration steps, will be displayed:

1. **Change the default administration password on the System/Administration page** (*Chapter 3*).
2. **Configure the local network settings on the System/IP page** (*Chapter 3*).
3. **Configure port settings and enable ….. the Serial & Network/Serial Port page** (*Chapter 4*).
4. **Configure users with access to serial ports on the Serial & Network/Users page** (*Chapter 3*).

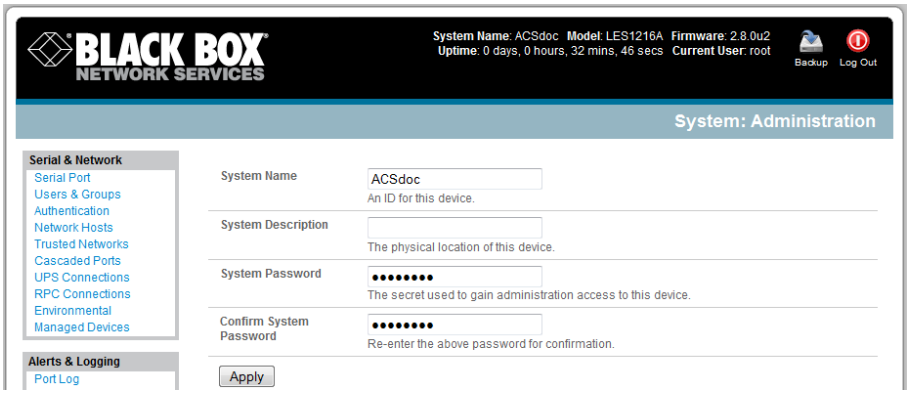
After completing each of the above steps, you can return to the configuration list by clicking in the top left corner of the screen on the Black Box logo.

Note If you are not able to connect to the Management Console at 192.168.0.1 or if the default Username/Password were not accepted, then reset your *console server* (refer to *Chapter 11*).

3.2 *Administrator* Password

For security reasons, only the administrator user named **root** can initially log into your *console server*. Only people who know the root password can access and reconfigure the *console server* itself. However, anyone who correctly guesses the root password could gain access (and the default root password is **default**). To avoid this, enter and confirm a new root password before giving the *console server* any access to, or control of, your computers and network appliances.

Note: We recommend that you set up a new *Administrator* user as soon as convenient and log in as this new user for all ongoing administration functions (rather than *root*). This *Administrator* can be configured in the *admin* group with full access privileges through the **Serial & Network: Users & Groups** menu as detailed in Ch*apter 4.*



- Select **System: Administration.**
- Enter a new **System Password** then re-enter it in **Confirm System Password.** This is the new password for **root**, the main administrative user account, so choose a complex password, and keep it safe.
- At this stage, you may also wish to enter a **System Name** and **System Description** for the *console server* to give it a unique ID and make it simple to identify.

Note The System Name can contain from 1 to 64 alphanumeric characters (however you can also use the special characters "-", "_", and ".")

There are no restrictions on the characters that can be used in the System Description or the System Password (each can contain up to 254 characters). However, only the first eight System Password characters are used to make the *password hash*.

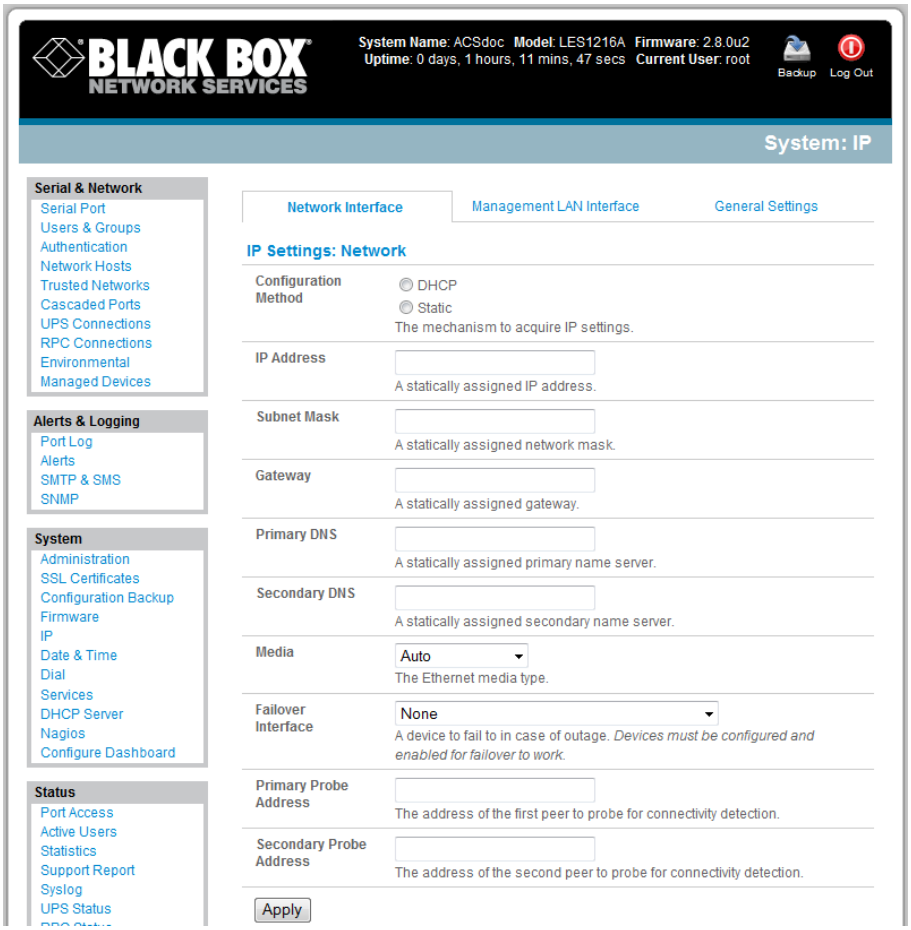
- Click **Apply**. Since you have changed the password you will be prompted to log in again. This time, use the new password.

Note If you are not confident that your *console server* has the current firmware release, you can upgrade. Refer to *Upgrade Firmware—Chapter 10*.

3.3 Network IP address

The next step is to enter an IP address for the principal Ethernet (*LAN/Network/Network1*) port on the *console server*; or enable its DHCP client so that it automatically obtains an IP address from a DHCP server on the network it will connect to.

- On the **System: IP** menu, select the **Network Interface** page then check **dhcp** or **static** for the **Configuration Method.**
- If you selected **Static,** you must manually enter the new **IP Address**, **Subnet Mask**, **Gateway,** and **DNS** server details. This selection automatically disables the DHCP client.



- If you selected **DHCP,** the *console server* will look for configuration details from a DHCP server on your management LAN. This selection automatically disables any static address. The *console server* MAC address is printed on a label on the base plate.

Note In its factory default state (with no Configuration Method selected) the *console server* has its DHCP client enabled, so it automatically accepts any network IP address assigned by a DHCP server on your network. In this initial state, the *console server* will then respond to both its Static address (192.168.0.1) and its newly assigned DHCP address.

- By default the *console server* LAN port auto-detects the Ethernet connection speed. You can use the **Media** menu to lock the Ethernet to 10 Mbps or 100 Mbps, and to Full Duplex (FD) or Half Duplex (HD).

Note If you changed the *console server* IP address, you may need to reconfigure your PC/workstation so it has an IP address that is in the same network range as this new address.

- Click **Apply.**
- Enter **http://new IP address** to reconnect the browser on the PC/workstation that is connected to the *console server*.

3.3.1 IPv6 configuration

You can also configure the *console server* Network and Management LAN Interfaces for IPv6 operation:

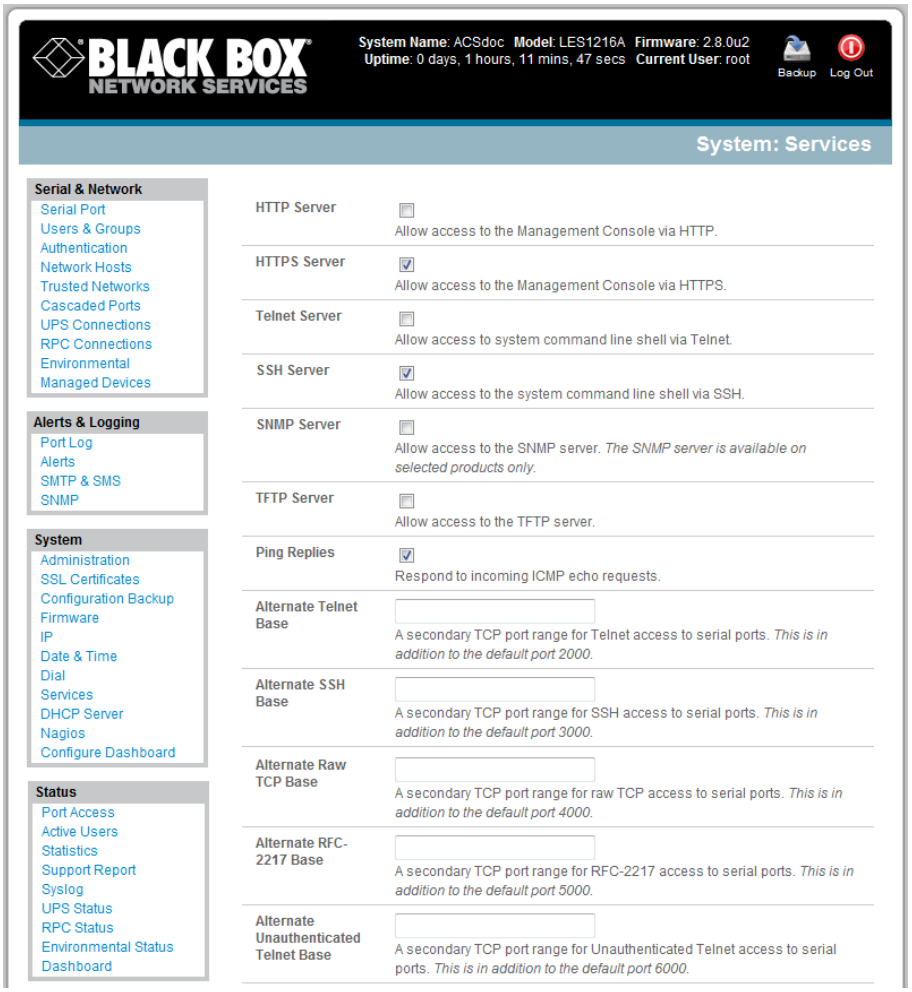
- On the **System: IP** menu select **General Settings** page and check **Enable IPv6.**
- Then, configure the IPv6 parameters on each Interface page.

3.4 System Services

The *Administrator* can access and configure the *console server* and connect to the managed devices using a range of access protocols (services). The factory default enables HTTPS and SSH access to the *console server* and disables HTTP and Telnet.

A *User* or *Administrator* can also use nominated enabled services to connect through the *console server* to attached serial and network connected managed devices.

The *Administrator* can simply disable any of the services, or enable others:


BLACK BOX NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 1 hours, 11 mins, 47 secs Current User: root
Backup
Log Out
System: Services
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Environmental
Managed Devices
Alerts & Logging
Port Log
Alerts
SMTP & SMS
SNMP
System
Administration
SSL Certificates
Configuration Backup
Firmware
IP
Date & Time
Dial
Services
DHCP Server
Nagios
Configure Dashboard
Status
Port Access
Active Users
Statistics
Support Report
Syslog
UPS Status
RPC Status
Environmental Status
Dashboard
HTTP Server
Allow access to the Management Console via HTTP.
HTTPS Server
Allow access to the Management Console via HTTPS.
Telnet Server
Allow access to system command line shell via Telnet.
SSH Server
Allow access to the system command line shell via SSH.
SNMP Server
Allow access to the SNMP server. The SNMP server is available on selected products only.
TFTP Server
Allow access to the TFTP server.
Ping Replies
Respond to incoming ICMP echo requests.
Alternate Telnet Base
A secondary TCP port range for Telnet access to serial ports. This is in addition to the default port 2000.
Alternate SSH Base
A secondary TCP port range for SSH access to serial ports. This is in addition to the default port 3000.
Alternate Raw TCP Base
A secondary TCP port range for raw TCP access to serial ports. This is in addition to the default port 4000.
Alternate RFC-2217 Base
A secondary TCP port range for RFC-2217 access to serial ports. This is in addition to the default port 5000.
Alternate Unauthenticated Telnet Base
A secondary TCP port range for Unauthenticated Telnet access to serial ports. This is in addition to the default port 6000.

- Select the **System: Services** option, then select/deselect for the service to be enabled/disabled. The following access protocol options are available:

  HTTPS This ensures secure browser access to all the Management Console menus. It also allows appropriately configured *Users* secure browser access to selected Management Console *Manage* menus. If you enable HTTPS, the *Administrator* will be able to use a secure browser connection to the *Console server*'s Management Console. For information on certificate and user client software configuration, refer to *Chapter 9—Authentication*. By default, HTTPS is enabled, and we recommend that that you only use HTTPS access if the *console server* will be managed over any public network (for example, the Internet).

  HTTP By default HTTP is disabled. We recommend that the HTTP service remain disabled if the *console server* will be remotely accessed over the Internet.

  Telnet This gives the *Administrator* Telnet access to the system command line shell (Linux commands). This may be suitable for a local direct connection over a management LAN. By default, Telnet is disabled. We recommend that this service remain disabled if you will remotely administer the *console server*.

  SSH This service provides secure SSH access to the Linux command line shell. We recommend that you choose SSH as the protocol where the *Administrator* connects to the *console server* over the Internet or any other public network. This will provide authenticated communications between the SSH client program on the remote PC/workstation and the SSH sever in the *console server*. By default SSH is enabled. For more information on SSH configuration refer *Chapter 9—Authentication*.

- You can configure related service options at this stage:

  SNMP This will enable *netsnmp* in the *console server*, which will keep a remote log of all posted information. SNMP is disabled by default. This SNMP service is only available in rackmount models. To modify the default SNMP settings, the *Administrator* must make the edits at the command line as described in *Chapter 15—Advanced Configuration.*

  TFTP This service will set up the default *tftp* server on the USB flash card (and is relevant to LES1208A, LES1216A and LES1248A *console server*s only). This server can be used to store config files, and maintain access and transaction logs, etc.

  Ping This allows the *console server* to respond to incoming ICMP echo requests. Ping is enabled by default. For security reasons, you should disable this service after initial configuration.

- And there are some serial port access parameters that you can configure on this menu:

  Base The *console server* uses specific default ranges for the TCP/IP ports for the various access services that *Users* and *Administrators* can use to access devices attached to serial ports (as covered in *Chapter 4—Configuring Serial Ports*). The *Administrator* can also set alternate ranges for these services, and these secondary ports will then be used in addition to the defaults.

  The default TCP/IP **base** port address for *telnet* access is 2000, and the range for *telnet* is IP Address: Port (2000 + serial port #) *i.e.* 2001 – 2048. If the *Administrator*

sets 8000 as a secondary base for telnet, then serial port #2 on the *console server* can be accessed via telnet at IP Address:2002 and at IP Address:8002.

The default base for SSH is 3000; for Raw TCP is 4000; and for RFC2217 it is 5000.

- Click **Apply**. As you apply your services selections, the screen will be updated with a confirmation message:

  Message Changes to configuration succeeded.

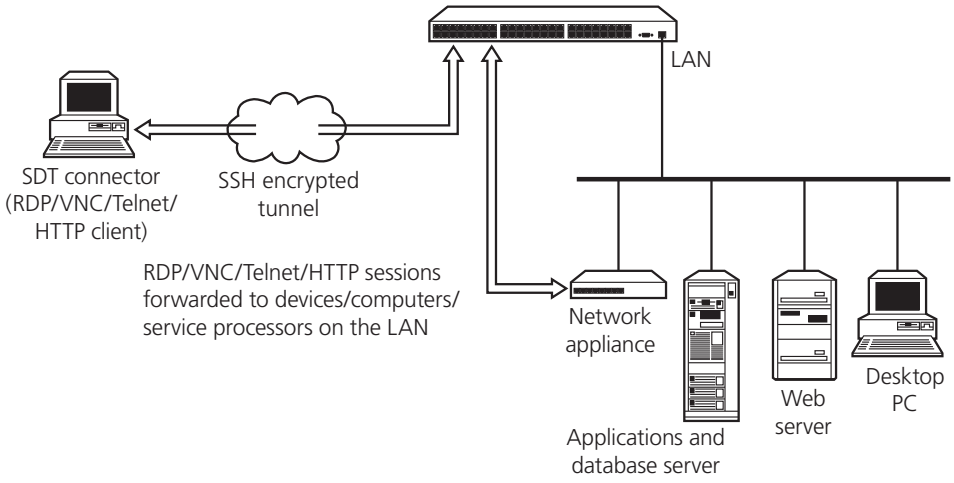
3.5 Communications Software

You have configured access protocols for the *Administrator* client to use when connecting to the *console server*. *User* clients (who you may set up later) will also use these protocols when accessing *console server* serial attached devices and network attached hosts. You will need to have appropriate communications software tools set up on the *Administrator* (and *User*) PC/workstation.

Black Box provides the *SDT Connector* Java applet as the recommended client software tool. You can use other generic tools such as PuTTY and SSHTerm. These tools are all described below as well.

3.5.1 SDT Connector

Each *console server* has an unlimited number of *SDT Connector* licenses to use with that *console server*.



SDT Connector is a lightweight tool that enables *Users* and *Administrators* to securely access the *console server* and the various computers, network devices, and appliances that may be serially or network connected to the *console server*.

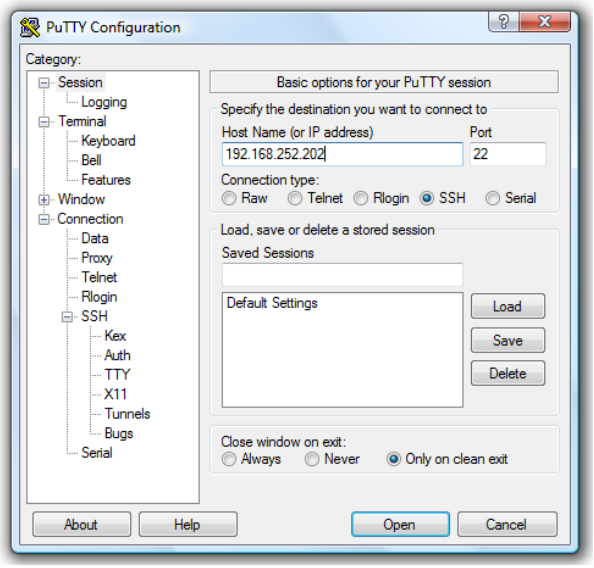
SDT Connector is a Java applet that couples the trusted SSH tunneling protocol with popular access tools such as Telnet, SSH, HTTP, HTTPS, VNC, and RDP to provide point-and-click secure remote management access to all the systems and devices being managed.

Information on using *SDT Connector* for browser access to the *console server*'s Management Console, Telnet/SSH access to the *console server* command line, and TCP/UDP connecting to hosts that are network connected to the *console server* is in *Chapter 6—Secure Tunneling.*

SDT Connector can be installed on Windows 2000, XP, 2003, Vista PCs, and on most Linux, UNIX, and Solaris computers.

3.5.2 PuTTY

You can also use communications packages like *PuTTY* to connect to the *console server* command line (and to connect serially attached devices as covered in *Chapter 4*). *PuTTY* is a freeware implementation of Telnet and SSH for Windows and UNIX platforms. It runs as an executable application without needing to be installed onto your system. *PuTTY* (the Telnet and SSH client itself) can be downloaded from *http://www.tucows.com/preview/195286.html*

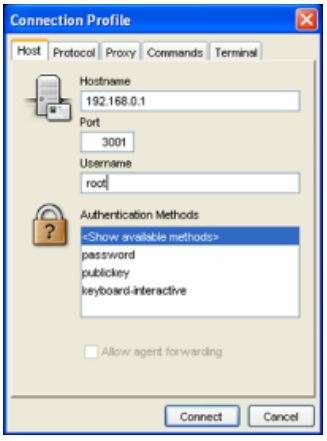


- To use PuTTY for an SSH terminal session from a Windows client, enter the *console server*'s IP address as the "Host Name (or IP address)."
- To access the *console server* command line, select "SSH" as the protocol, and use the default IP Port 22.
- Click "Open" and the *console server* login prompt will appear. (You may also receive a "Security Alert" that the host's key is not cached. Choose "yes" to continue.)
- Using the Telnet protocol is similarly simple - but you use the default port 23.

3.5.3 SSHTerm

Another popular communications package you can use is *SSHTerm,* an open source package that you can download from *http://sourceforge.net/projects/sshtools*

- To use *SSHTerm* for an SSH terminal session from a Windows Client, simply Select the "File" option and click on "New Connection."



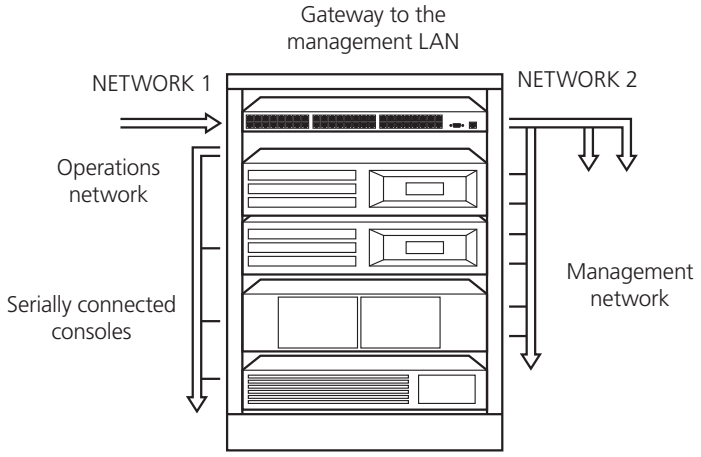
- A new dialog box will appear for your "Connection Profile." Type in the host name or IP address (for the *console server* unit) and the TCP port that the SSH session will use (port 22). Then type in your username, choose password authentication, and click connect.
- You may receive a message about the host key fingerprint. Select "yes" or "always" to continue.
- The next step is password authentication. The system prompts you for your username and password from the remote system. This logs you on to the *console server*

3.6 Management network configuration (LES1208A, LES1216A and LES1248A only)

The LES1208A, LES1216A, and LES1248A *console server*s have a second network port that you can configure as a management LAN port or as a failover/ OOB access port.

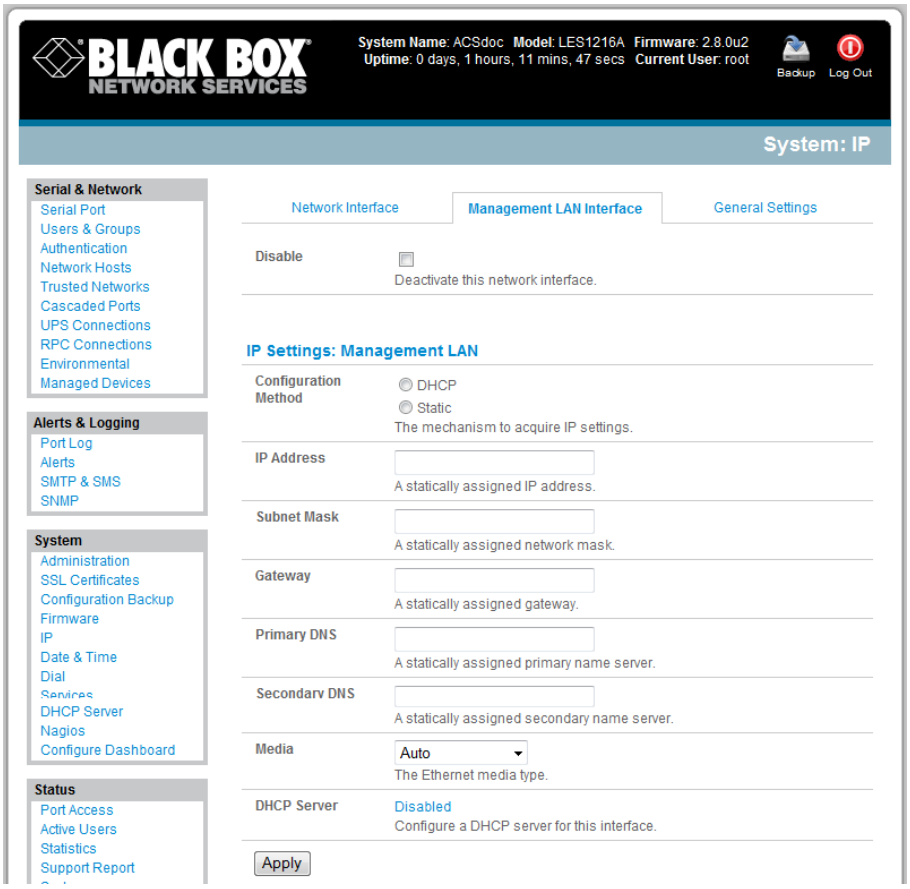
3.6.1 Enable the Management LAN

The LES1208A, LES1216A, and LES1248A *console server*s provide a firewall, router, and DHCP server. You need to connect an external LAN switch to *Network 2* to attach hosts to this management LAN.



This Management LAN feature is disabled by default. To configure the Management LAN gateway:

- Select the **Management LAN** page on the **System: IP** menu and uncheck **Disable.**
- Configure the **IP Address** and **Subnet Mask** for the Management LAN (but leave the **DNS** fields blank).
- Click **Apply.**



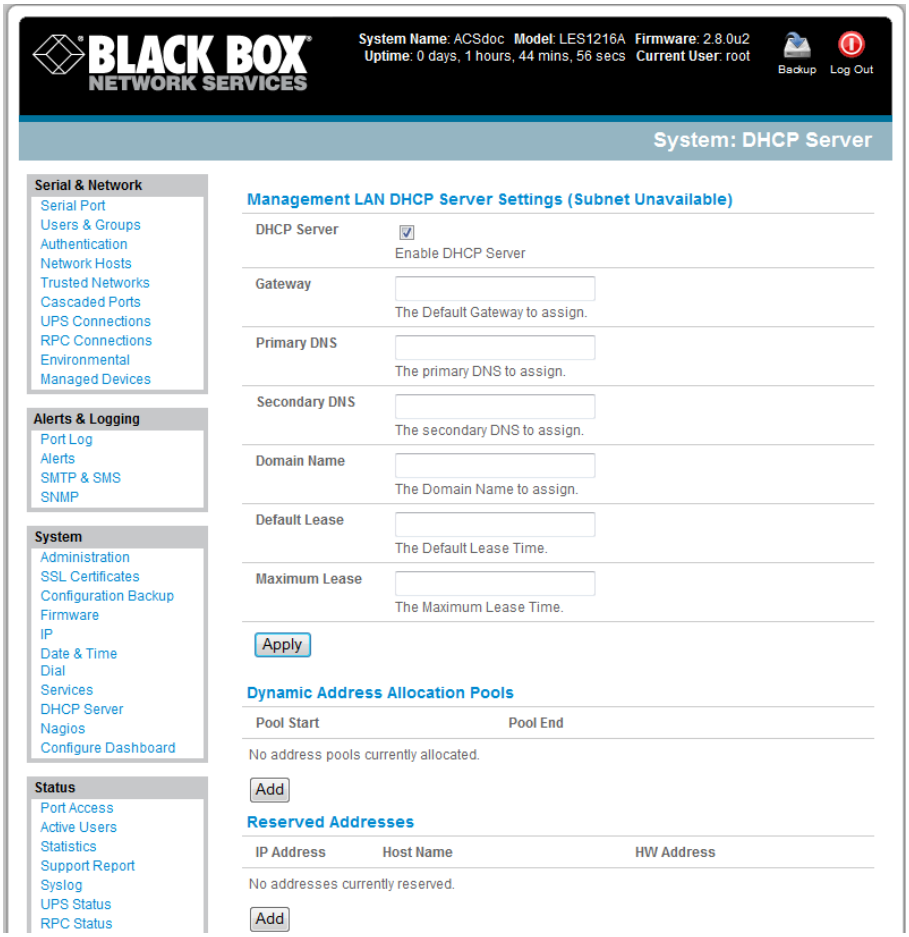
Note You can configure the second Ethernet port as either a gateway port or as an OOB/Failover port (but not both). Make sure you did not allocate **Network 2** as the **Failover Interface** when you configured the principal **Network** connection on the **System: IP** menu.

The management gateway function is now enabled with default firewall and router rules. By default, these rules are configured so the Management LAN can only be accessible by SSH port forwarding. This ensures that the remote and local connections to Managed Devices on the Management LAN are secure. You can also configure the LAN ports in bridged mode (as described later in this chapter) or you can configure them from the command line.

3.6.2 Configure the DHCP server

The LES1208A, LES1216A, and LES1248A *console server*s also host a DHCP server which by default is disabled. The DHCP server enables the automatic distribution of IP addresses to hosts on the Management LAN that are running DHCP clients. To enable the DHCP server:

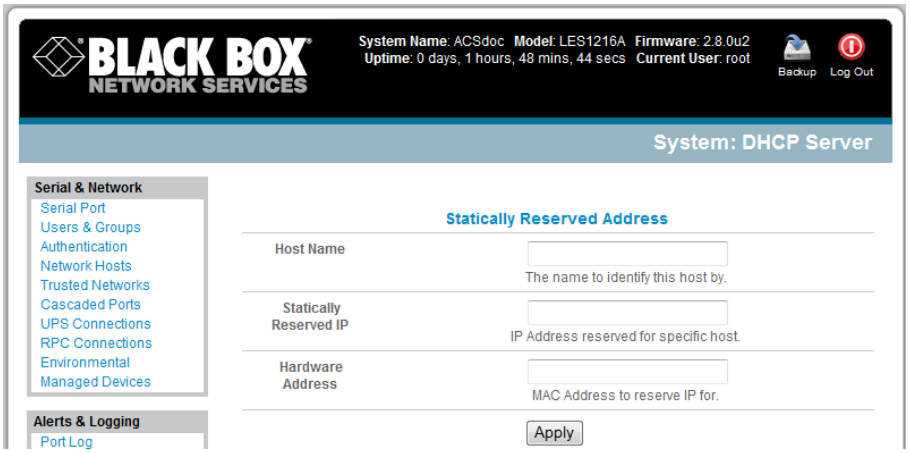
- On the **System: IP** menu select the **Management LAN** page and click the **Disable** label in the **DHCP Server** field (or go to the **System: DHCP Server** menu and check **Enable DHCP Server**).



- Enter the **Gateway** address that you want to issue to the DHCP clients. If you leave this field blank, the *console server*'s IP address will be used.
- Enter the **Primary DNS** and **Secondary DNS** address to issue the DHCP clients. If you leave this field blank, the *console server*'s IP address is used. So, leave this field blank for automatic DNS server assignment.
- Optionally, enter a **Domain Name** suffix to issue DHCP clients.
- Enter the **Default Lease** time and **Maximum Lease** time in seconds. The lease time is the time that a dynamically assigned IP address is valid before the client must request it again.
- Click **Apply.**

The DHCP server will sequentially issue IP addresses from a specified address pool(s):

- Click **Add** in the **Dynamic Address Allocation Pools** field.
- Enter the **DHCP Pool Start Address** and **End Address** and click **Apply.**



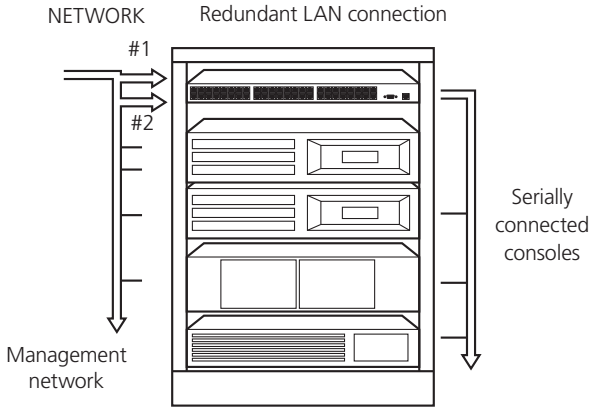
The DHCP server also supports pre-assigning IP addresses to be allocated only to specific MAC addresses and reserving IP addresses to be used by connected hosts with fixed IP addresses. To reserve an IP addresses for a particular host:

- Click **Add** in the **Reserved Addresses** field.
- Enter the **Hostname,** the **Hardware Address** (MAC), and the **Statically Reserved IP** address for the DHCP client and click **Apply**.

When DHCP has initially allocated hosts addresses, copy these addresses into the pre-assigned list so the same IP address will be reallocated if you reboot the system.

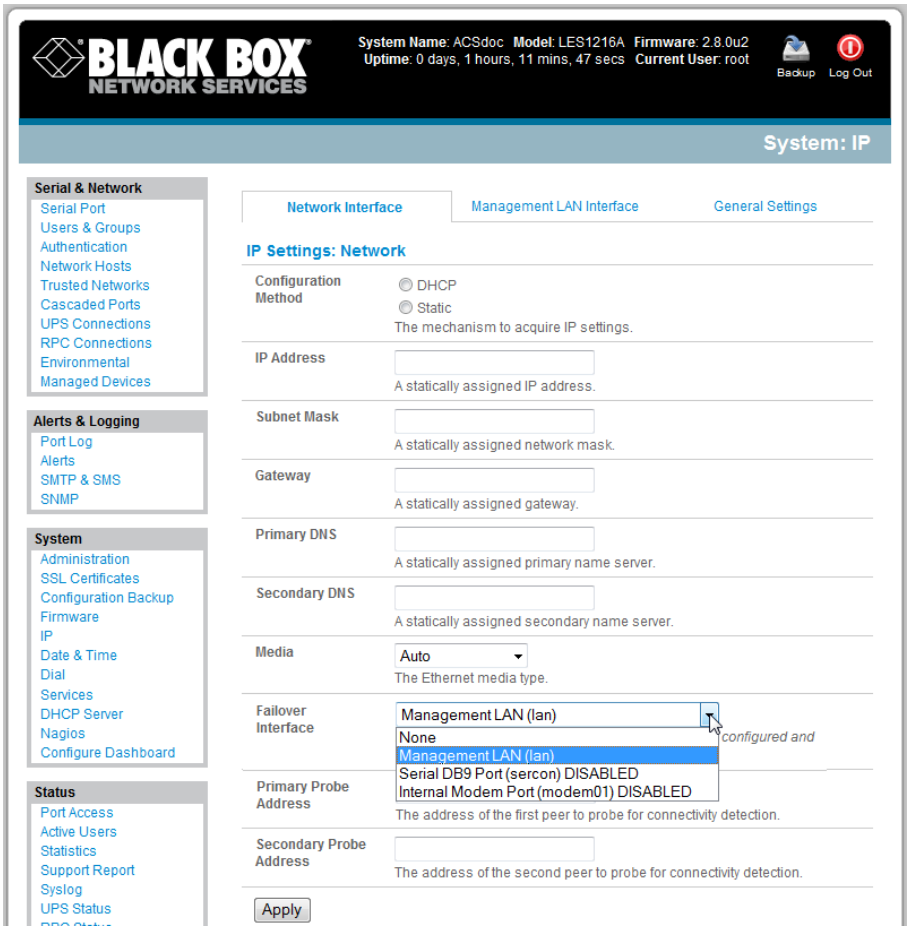
3.6.3 Select Failover or broadband OOB

The LES1208A, LES1216A and LES1248A *console server*s provide a broadband failover option. If you have a problem using the main LAN connection for accessing the *console server*, an alternate access path is used.



- By default, the failover is not enabled. To enable, select the **Network** page on the **System: IP** menu.
- Select the **Failover Interface** to be used if the main fails. This can be:
    - an alternate broadband Ethernet connection (which would be the *Network2* port on the LES1208A, LES1216A, and LES1248A) or

- the internal modem, or
- an external serial modem connected to the Console port (for dialing out to an ISP or the remote management office).

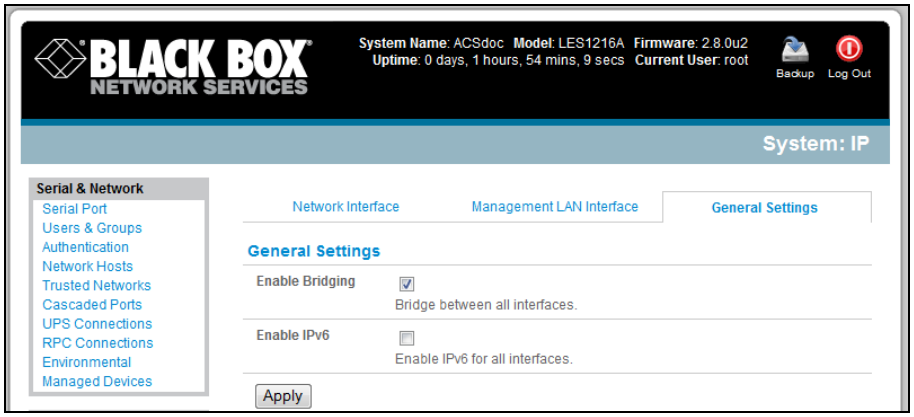


➢ Click **Apply**. You have selected the failover method. It is not active until you specify the external sites to be probed to trigger failover, and set up the failover ports themselves. This is covered in *Chapter 5*.

Note With the LES1208A, LES1216A, and LES1248A, you can configure the second Ethernet port as either a gateway port or as an OOB/Failover port, but not both. Make sure you did not enable the Management LAN function on **Network 2.**

3.6.4 Bridging the network ports

By default, you can only access the *console server*'s Management LAN network ports using SSH tunneling/port forwarding. However, all the wired network ports on the *console server*s can also be bridged.



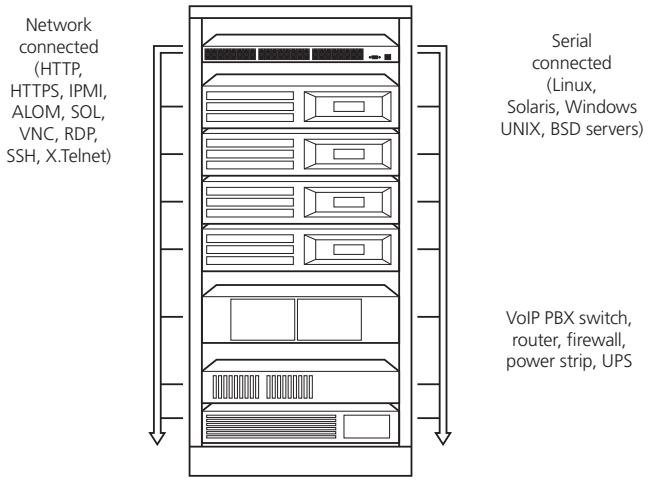
- Select **Enable Bridging** on the **System: IP General Settings** menu.
- All the Ethernet ports are all transparently connected at the data link layer (layer 2) and they are configured collectively using the **Network Interface** menu.

When bridging is enabled, network traffic is forwarded between all Ethernet ports with no firewall restrictions. This mode also removes all the **Management LAN Interface** and **Out-of-Band/Failover Interface** functions, and disables the **DHCP Server**.

Chapter 4 Serial Port, Host, Device & User Configuration

Introduction

The Black Box *console server* enables access and control of serially attached devices and network attached devices (*hosts*). The *Administrator* must configure access privileges for each of these devices, and specify the services that can be used to control the devices. The *Administrator* can also set up new users and specify each user's individual access and control privileges.


Network connected (HTTP, HTTPS, IPMI, ALOM, SOL, VNC, RDP, SSH, X.Telnet)
Serial connected (Linux, Solaris, Windows UNIX, BSD servers)
VoIP PBX switch, router, firewall, power strip, UPS

This chapter covers each of the steps in configuring hosts and serially attached devices:

Configure Serial Ports—setting up the protocols to be used in accessing serially-connected devices.
Users & Groups—setting up users and defining the access permissions for each of these users.
Authentication—covered in more detail in Chapter 9.
Network Hosts—configuring access to network connected devices (referred to as hosts).
Configuring Trusted Networks—nominate user IP addresses.
Cascading and Redirection of Serial Console Ports.
Connecting to Power (UPS PDU and IPMI) and Environmental Monitoring (EMD) devices.
Managed Devices—presents a consolidted view of all the connections.

4.1 Configure Serial Ports

To configure a serial port, you must first set the **Common Settings** (the protocols and the RS-232 parameters (such as baud rate) that will be used for the data connection to that port.

Select what mode the port is to operate in. You can set each port to support one of five operating modes:

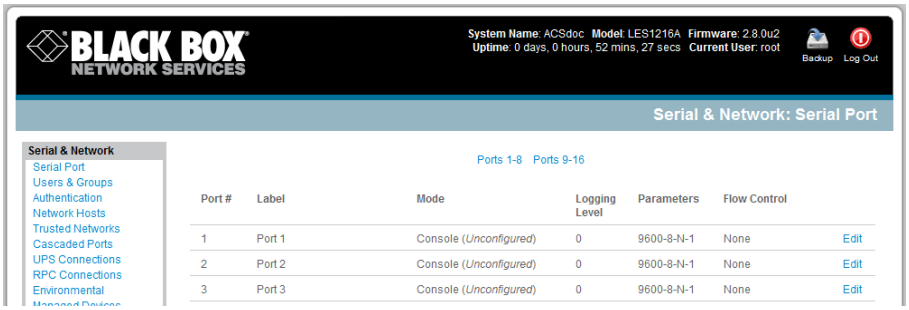
1) Console Server Mode is the default and this enables general access to serial console port on the serially attached devices.

2) Device Mode sets the serial port up to communicate with an intelligent serial controlled PDU, UPS, or Environmental Monitor Device (EMD).

3) SDT Mode enables graphical console access (with RDP, VNC, HTTPS, etc.) to hosts that are serially connected.

4) Terminal Server Mode sets the serial port to wait for an incoming terminal login session.

5) Serial Bridge Mode enables transparently interconnects two serial port devices over a network.


BLACK BOX
NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 0 hours, 52 mins, 27 secs Current User: root
Backup
Log Out
Serial & Network: Serial Port
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Environmental
Ports 1-8 Ports 9-16
Port # Label Mode Logging Level Parameters Flow Control
1 Port 1 Console (Unconfigured) 0 9600-8-N-1 None Edit
2 Port 2 Console (Unconfigured) 0 9600-8-N-1 None Edit
3 Port 3 Console (Unconfigured) 0 9600-8-N-1 None Edit

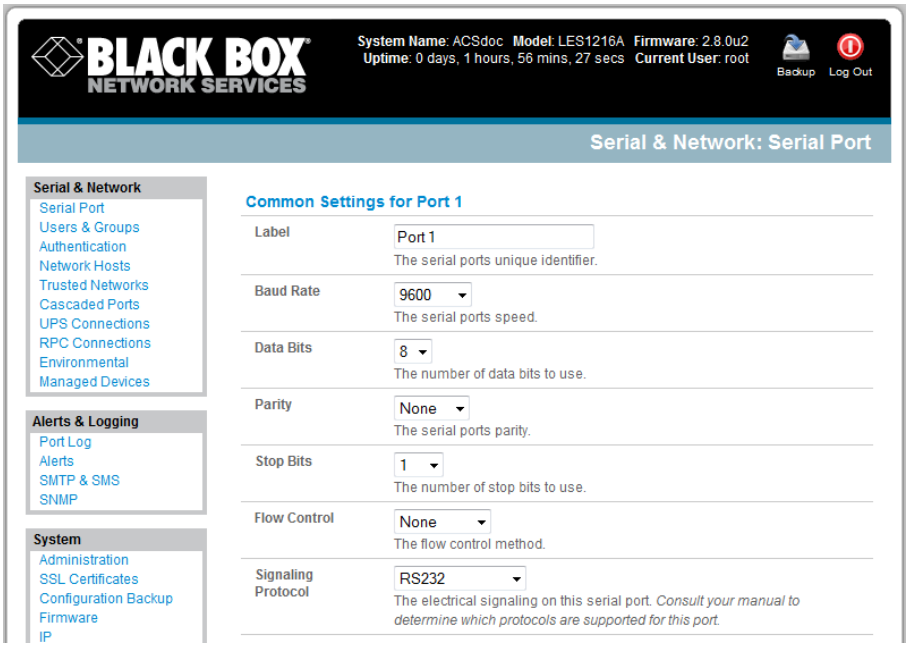
- Select **Serial & Network: Serial Port** and you will see the current labels, modes, logging levels, and RS-232 protocol options that are currently set up for each serial port.
- By default, each serial port is set in Console Server mode. To reconfigure the port, click **Edit**.
- When you have reconfigured the common settings (*Chapter 4.1.1*) and the mode (*Chapters 4.1.2 –4.1.6*) for each port, you can set up any remote syslog (*Chapter 4.1.7*), then click **Apply.**

Note If you want to set the same protocol options for multiple serial ports at once, click **Edit Multiple Ports** and select which ports you want to configure as a group.

- If the *console server* has been configured with distributed Nagios monitoring enabled, then you will also be presented with **Nagios Settings** options to enable nominated services on the Host to be monitored (refer *Chapter 10—Nagios Integration*).

4.1.1 Common Settings

There are a number of common settings that you can set for each serial port. These are independent of the mode in which the port is being used. Set these serial port parameters to match the serial port parameters on the device you attach to that port.

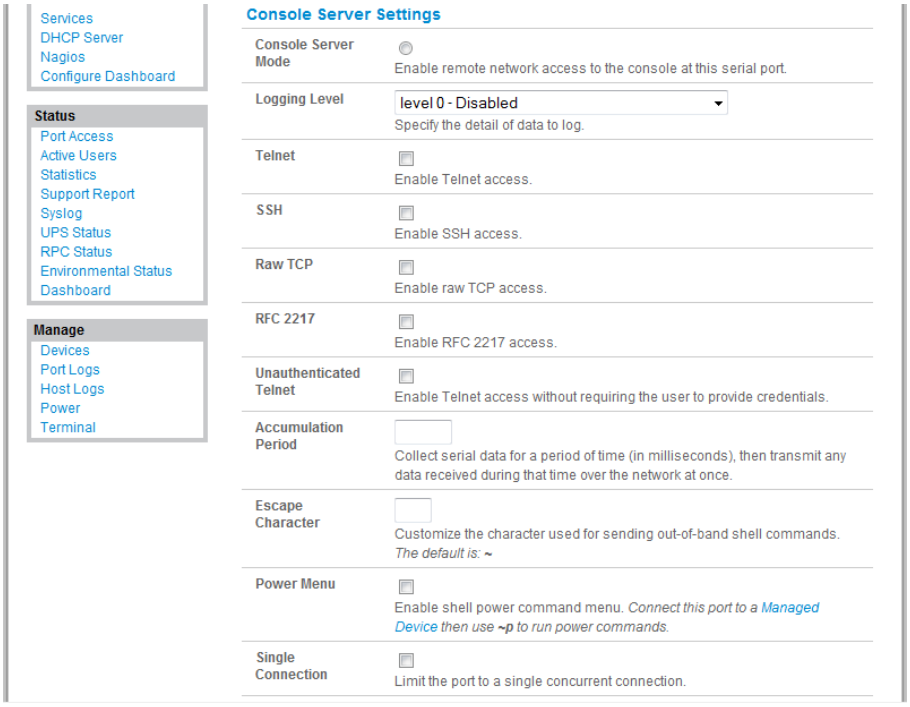


- Specify a label for the port.
- Select the appropriate **Baud Rate**, **Parity**, **Data Bits**, **Stop Bits,** and **Flow Control** for each port. (Note: The RS-485/RS-422 option is not relevant for *console server*s.)
- Before proceeding with further serial port configuration, connect the ports to the serial devices they will be controlling, and make sure they have matching settings.

Note The serial ports are all set at the factory to RS232 9600 baud, no parity, 8 data bits, 1 stop bit, and *Console server* Mode. You can change the baud rate to 2400–230400 baud using the management console. You can configure lower baud rates (50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800 baud) from the command line. Refer to *Chapter 14— Basic Configuration (Linux Commands).*

4.1.2 Console Server Mode

Select **Console Server Mode** to enable remote management access to the serial console that is attached to this serial port:

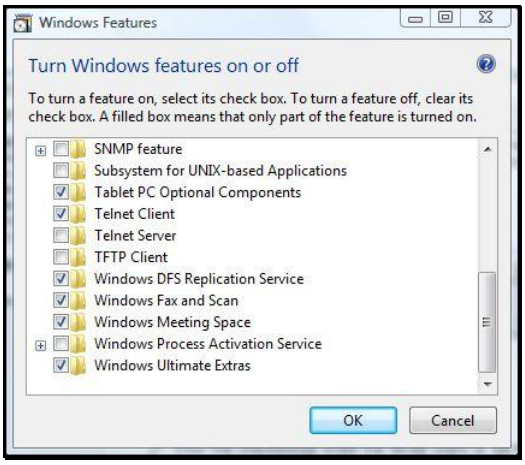


Logging Level This specifies the level of information to be logged and monitored (referto *Chapter 7—Alerts and Logging).*

Telnet When the Telnet service is enabled on the *console server*, a Telnet client on a *User* or *Administrator's* computer can connect to a serial device attached to this serial port on the *console server*. The Telnet communications are unencrypted, so this protocol is generally recommended only for local connections.

With Win2000/XP/NT you can run *telnet* from the command prompt (*cmd.exe*). Vista and Windows 7 include a Telnet client and server, but they are not enabled by default. To enable Telnet:

- Log in as *Admin* and go to *Start/Control Panel/Programs and Features.*
- Select *Turn Windows features on or off,* check the *Telnet Client,* and click *OK.*


Windows Features
Turn Windows features on or off
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.
SNMP feature
Subsystem for UNIX-based Applications
Tablet PC Optional Components
Telnet Client
Telnet Server
TFTP Client
Windows DFS Replication Service
Windows Fax and Scan
Windows Meeting Space
Windows Process Activation Service
Windows Ultimate Extras
OK
Cancel

If the remote communications are tunneled with *SDT Connector,* then you can use Telnet to securely access these attached devices (refer to the Note below).

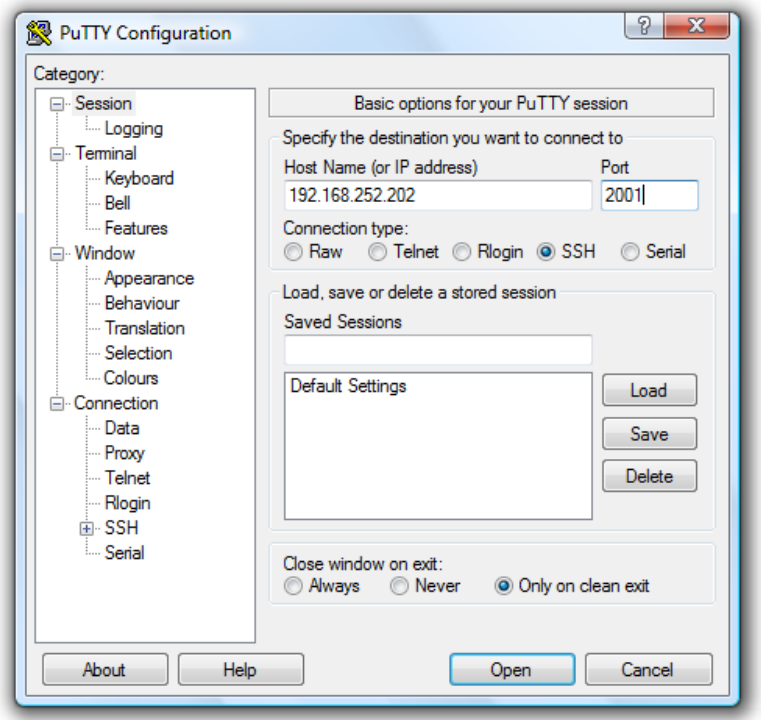
Note In Console Server mode, *Users* and *Administrators* can use *SDT Connector* to set up secure Telnet connections that are SSH tunneled from their client PC/workstations to the serial port on the *console server*. *SDT Connector* can be installed on Windows 2000, XP, 2003, Vista, and Windows 7 PCs and on most Linux platforms. You can also set up secure Telnet connections with a simple point-and-click.

To use *SDT Connector* to access consoles on the *console server* serial ports, you configure *SDT Connector* with the *console server* as a *gateway*, then configure it as a *host*, Next, you enable Telnet service on Port (2000 + serial port #) *i.e.* 2001–2048. Refer to *Chapter 6* for more details on using *SDT Connector* for Telnet and SSH access to devices that are attached to the *console server* serial ports.

You can also use standard communications packages like *PuTTY* to set a direct Telnet (or SSH) connection to the serial ports (refer to the Note below).

Note *PuTTY* also supports Telnet (and SSH) and the procedure to set up a Telnet session is simple. Enter the *console server's* IP address as the "Host Name (or IP address)." Select "Telnet" as the protocol and set the "TCP port" to 2000 plus the physical serial port number (*that is,* 2001 to 2048).

Click the "Open" button. You may then receive a "Security Alert" that the host's key is not cached. Choose "yes" to continue. You will then be presented with the login prompt of the remote system connected to the serial port chosen on the *console server*. Login as normal and use the host serial console screen.



PuTTY can be downloaded at http://www.tucows.com/preview/195286.html

SSH

We recommend that you use SSH as the protocol where the *User* or *Administrator* connects to the *console server* (or connects through the *console server* to the attached serial consoles) over the Internet or any other public network. This will provide authenticated SSH communications between the SSH client program on the remote user's computer and the *console server*, so the user's communication with the serial device attached to the *console server* is secure.

For SSH access to the consoles on devices attached to the *console server* serial ports, you can use *SDT Connector*. Configure *SDT Connector* with the *console server* as a *gateway*, then as a *host*, and enable SSH service on Port (3000 + serial port #) *i.e.* 3001-3048. *Chapter 6—Secure Tunneling* has more information on using *SDT Connector* for SSH access to devices that are attached to the *console server* serial ports.

You can also use common communications packages, like *PuTTY* or *SSHTerm* to SSH connect directly to port address IP Address _ Port (3000 + serial port #) *i.e.* 3001–3048.

SSH connections can be configured using the standard SSH port 22. Identify the the serial port that's accessed by appending a descriptor to the username. This syntax supports:

<username>:<portXX>

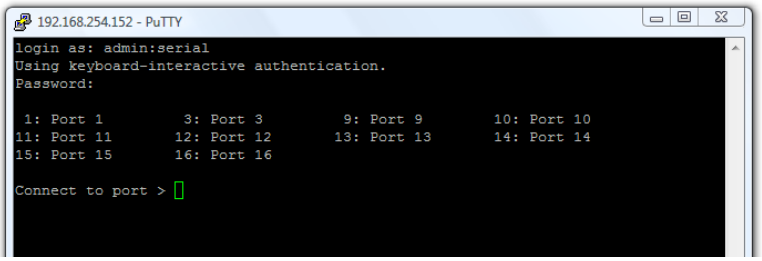
<username>:<port label>

<username>:<ttySX>

<username>:<serial>

For a *User* named “fred” to access serial port 2, when setting up the SSHTerm or the PuTTY SSH client, instead of typing *username = fred* and *ssh port = 3002*, the alternate is to type *username = fred:port02 (*or *username = fred:ttyS1)* and *ssh port = 22.*

Or, by typing *username=fred:serial* and *ssh port = 22.* A port selection option appears to the *User*:


192.168.254.152 - PuTTY
login as: admin:serial
Using keyboard-interactive authentication.
Password:
1: Port 1 3: Port 3 9: Port 9 10: Port 10
11: Port 11 12: Port 12 13: Port 13 14: Port 14
15: Port 15 16: Port 16
Connect to port >

This syntax enables *Users* to set up SSH tunnels to all serial ports with only opening a single IP port 22 in their firewall/gateway.

TCP RAW TCP allows connections directly to a TCP socket. Communications programs like *PuTTY* also support RAW TCP. You would usually access this protocol via a custom application.

For RAW TCP, the default port address is IP Address _ Port (4000 + serial port #) *i.e.* 4001 – 4048.

RAW TCP also enables the serial port to be tunneled to a remote *console server*, so two serial port devices can transparently interconnect over a network (see *Chapter 4.1.6—Serial Bridging*).

RFC2217 Selecting *RFC2217* enables serial port redirection on that port. For RFC2217, the default port address is IP Address _ Port (5000 + serial port #), that is, 5001 – 5048.

Special client software is available for Windows UNIX and Linux that supports RFC2217 virtual com ports, so a remote host can monitor and manage remote serially attached devices, as though they were connected to the local serial port (see *Chapter 4.6—Serial Port Redirection* for details).

RFC2217 also enables the serial port to be tunneled to a remote *console server*, so two serial port devices can transparently interconnect over a network (see *Chapter 4.1.6—Serial Bridging*).

Unauthenticated Telnet Selecting *Unauthenticated Telnet* enables telnet access to the serial port without requiring the user to provide credentials. When a user accesses the *console server* to telnet to a serial port he normally is given a login prompt. With unauthenticated telnet, the user connects directly through to a port with any *console server* login. This mode is mainly used when you have an external system (such as *conserver*) managing user authentication and access privileges at the serial device level.

For Unauthenticated Telnet, the default port address is IP Address _ Port (6000 + serial port #) i.e. 6001 – 6048

Accumulation Period By default, once a connection is established for a particular serial port (such as a RFC2217 redirection or Telnet connection to a remote computer) then any incoming characters on that port are forwarded over the network on a character by character basis. The accumulation period changes this by specifying a period of time that incoming characters will be collected before then being sent as a packet over the network.

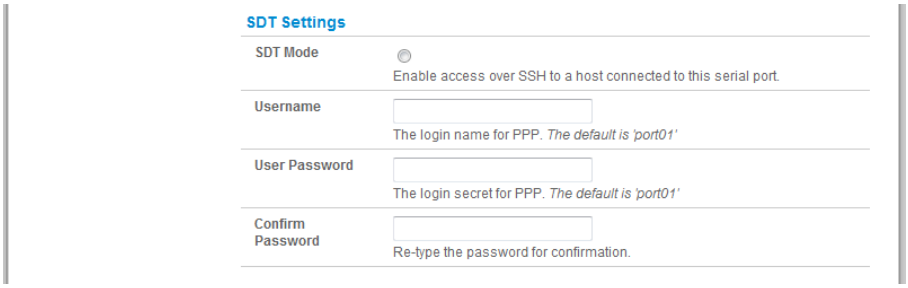
Escape Character This enables you to change the character used for sending escape characters. The default is ~.

Power Menu This setting enables the shell power command. A user can control the power connection to a Managed Device from command line when they are connected to the device via telnet or ssh. To operate, the Managed Device must be set up with both its Serial port connection and Power connection configured. The command to bring up the power menu is **~p**

Single Connection This setting limits the port to a single connection> If multiple users have access privileges for a particular port, only one user at a time can access that port (that is, port "snooping" is not permitted).

4.1.3 SDT Mode

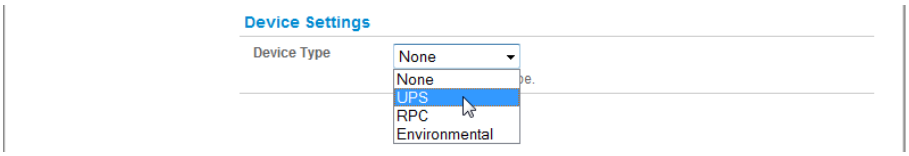
This setting allows port forwarding of RDP, VNC, HTPP, HTTPS, SSH, Telnet, and other LAN protocols through to computers that are locally connected to the *console server* by their serial COM port. Port forwarding requires that you set up a PPP link over this serial port.


SDT Settings
SDT Mode
Enable access over SSH to a host connected to this serial port.
Username
The login name for PPP. The default is 'port01'
User Password
The login secret for PPP. The default is 'port01'
Confirm Password
Re-type the password for confirmation.

For configuration details, refer to *Chapter 6.6—Using SDT Connector to Telnet or SSH connect to devices that are serially attached to the console server*.

4.1.4 Device (RPC, UPS, EMD) Mode

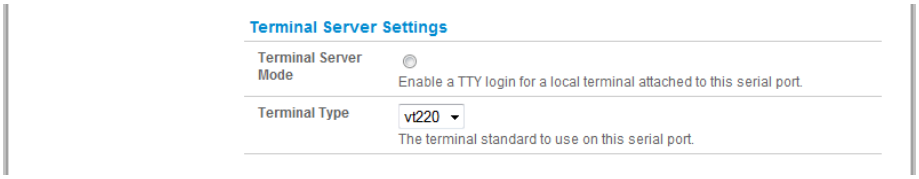
This mode configures the selected serial port to communicate with a serial controlled Uninterruptable Power Supply (UPS), Remote Power Controller/Power Distribution Unit (RPC) or Environmental Monitoring Device (EMD).



- Select the desired **Device Type** (UPS, RPC or EMD)
- Proceed to the appropriate device configuration page (**Serial & Network: UPS Connections, RPC Connection** or **Environmental**) as detailed in *Chapter 8—Power & Environmental Management.*

4.1.5 Terminal Server Mode

- Select **Terminal Server Mode** and the **Terminal Type** (vt220, vt102, vt100, Linux, or ANSI) to enable a *getty* on the selected serial port.


Terminal Server Settings
Terminal Server Mode
Enable a TTY login for a local terminal attached to this serial port.
Terminal Type
vt220
The terminal standard to use on this serial port.

The *getty* will then configure the port and wait for a connection to be made. An active connection on a serial device is usually indicated by the Data Carrier Detect (DCD) pin on the serial device being raised. When a connection is detected, the *getty* program issues a login: prompt, and then invokes the login program to handle the actual system login.

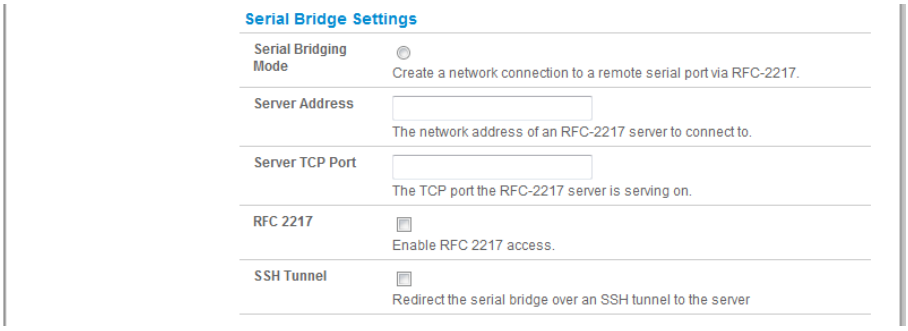
Note Selecting Terminal Server mode will disable Port Manager for that serial port, so data is no longer logged for alerts, etc.

4.1.6 Serial Bridging Mode

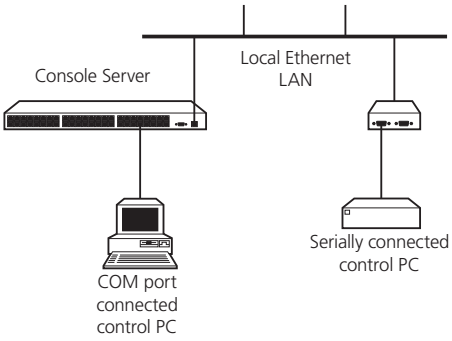
With serial bridging, the serial data on a nominated serial port on one *console server* is encapsulated into network packets and then transported over a network to a second *console server*. It is then represented on its serial port again as serial data. The two *console server*s effectively act as a virtual serial cable over an IP network.

One *console server* is configured as the *Server*. Set the *Server* serial port to be bridged in Console Server mode with either RFC2217 or RAW enabled (as described in *Chapter 4.1.2—Console Server Mode*).

For the *Client console server*, the serial port to bridge must be set in Bridging Mode:


Serial Bridge Settings
Serial Bridging Mode
Create a network connection to a remote serial port via RFC-2217.
Server Address
The network address of an RFC-2217 server to connect to.
Server TCP Port
The TCP port the RFC-2217 server is serving on.
RFC 2217
Enable RFC 2217 access.
SSH Tunnel
Redirect the serial bridge over an SSH tunnel to the server

- Select **Serial Bridging Mode** and specify the IP address of the *Server console server* and the TCP port address of the remote serial port (for RFC2217 bridging this will be 5001-5048).
- By default, the bridging client will use RAW TCP. Select RFC2217 if this is the *console server* mode you have specified on the server *console server.*

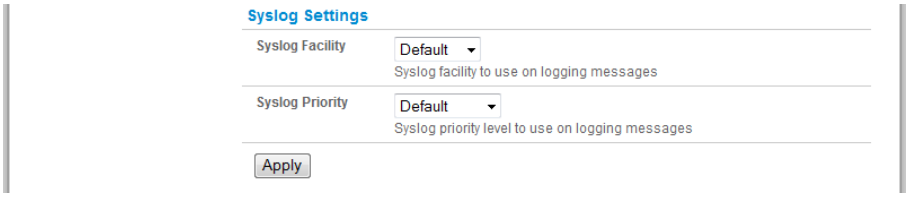


- You may secure the communications over the local Ethernet by enabling SSH. You will need to generate and upload keys (refer to *Chapter 14— Advanced Configuration*).

4.1.8 Syslog

In addition to built-in logging and monitoring (which can be applied to serial-attached and network-attached management accesses, as covered in *Chapter 7—Alerts and Logging*), you can also configure the *console server* to support the remote syslog protocol on a per serial port basis:

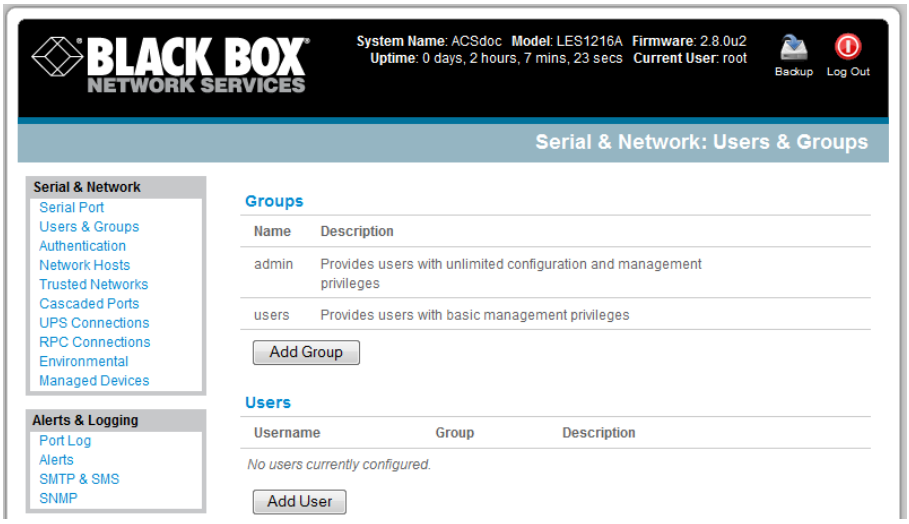
- Select the **Syslog Facility/Priority** fields to enable logging of traffic on the selected serial port to a syslog server; and to appropriately sort and action those logged messages (that is, redirect them/send alert email etc.).



For example, if the computer attached to serial port 3 should never send anything out on its serial console port, the *Administrator* can set the **Facility** for that port to *local0* (*local0 .. local7* are for site local values), and the **Priority** to *critical*. At this priority, if the *console server* syslog server does receive a message, it will automatically raise an alert. Refer to *Chapter 7—Alerts & Logging.*

4.2 Add/ Edit Users

The *Administrator* uses this menu selection to set up, edit, and delete users, and to define the access permissions for each of these users.



Users can be authorized to access specified *console server* serial ports and specified network-attached hosts. These users can also be given full *Administrator* status (with full configuration and management and access privileges).

To simplify user set up, they can be configured as members of Groups. There are two Groups set up by default (*admin* and *user*).

1. Members of the ***admin*** group have full *Administrator* privileges. The *admin* user (*Administrator*) can access the *console server* using any of the services that are enabled in *System: Services*. For example, if only HTTPS has been enabled, then the *Administrator* can only access the *console server* using HTTPS. Once logged in, they can reconfigure the *console server* settings (for example, to enabled HTTP/Telnet for future access). They can also access any of the connected Hosts or serial port devices using any of the services that have been enabled for these connections. The *Administrator* can reconfigure the access services for any Host or serial port. Only trusted users should have *Administrator* access.

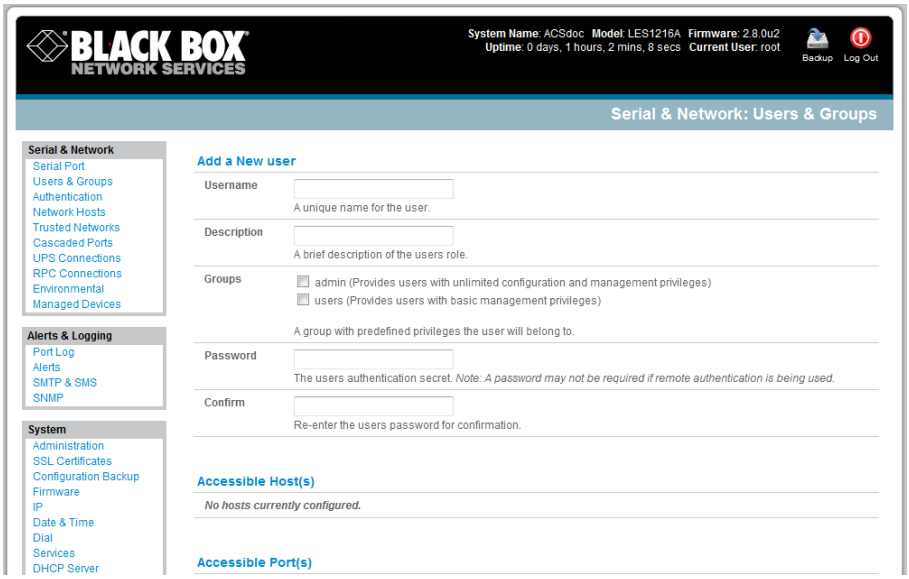
Note: For convenience, the SDT Connector "Retrieve Hosts" function retrieves and auto-configures checked serial ports and checked hosts only, even for admin group users.

2. Members of the **user** group have limited access to the *console server* and connected Hosts and serial devices. These *Users* can access only the Management section of the Management Console menu and they have no command line access to the *console server*. They also can only access those Hosts and serial devices that are checked for them, using services that are enabled.
3. The *Administrator* can also set up additional Groups with specific serial port and host access permissions (same as *Users*). However, users in these additional groups don't have any access to the Management Console menu or any command line access to the *console server* itself. Finally,

the *Administrator* can also set up users who are not a member of any Groups. They will have the same access as users in the additional groups.

To set up new Groups and new users, and to classify users as members of particular Groups:

- Select **Serial & Network: Users & Groups** to display the configured Groups and Users.
- Click **Add Group** to add a new Group.
- Add a **Group** name and **Description** for each new Group, then nominate the **Accessible Hosts, Accessible Ports,** and **Accessible RPC Outlets(s)** that you want any users in this new Group to be able to access.
- Click **Apply.**



- Click **Add User** to add a new user.
- Add a **Username** and a confirmed **Password** for each new user. You may also include information related to the user (for example, contact details) in the **Description** field.

Note The User Name can contain from 1 to 127 alphanumeric characters (you can also use the special characters "-", "_", and ".").

There are no restrictions on the characters that you can use in the user Password (each can contain up to 254 characters). Only the first eight Password characters are used to make the *password hash*.

- Specify which **Group** (or Groups) you want the user to join.
- Check specific **Accessible Hosts** and/or **Accessible Ports** to nominate the serial ports and network connected hosts you want the user to have access privileges to.
- If there are configured RPCs, you can check **Accessible RPC Outlets** to specify which outlets the user is able to control (that is, Power On/Off).

- Click **Apply**. The new user can now access the Network Devices, Ports, and RPC Outlets you nominated as accessible. Plus, if the user is a Group member they can also access any other device/port/outlet that was set up as accessible to the Group.

Note There are no specific limits on the number of users you can set up; nor on the number of users per serial port or host. Multiple users (*Users* and *Administrators*) can control/monitor one port or host.

There are no specific limits on the number of Groups. Each user can be a member of a number of Groups (they take on the cumulative access privileges of each of those Groups). A user does not have to be a member of any Groups (but if the *User* is not even a member of the default *user* group. then he will not be able to use the Management Console to manage ports).

The time allowed to re-configure increases as the number and complexity increases. We recommend that you keep the aggregate number of users and groups under 250.

The *Administrator* can also edit the access settings for any existing users:

- Select **Serial & Network: Users & Groups** and click **Edit** for the *User* to be modified.

Note For more information on enabling the SDT Connector so each user has secure tunneled remote RPD/VNC/Telnet/HHTP/HTTPS/SoL access to the network connected hosts, refer to *Chapter 6*.

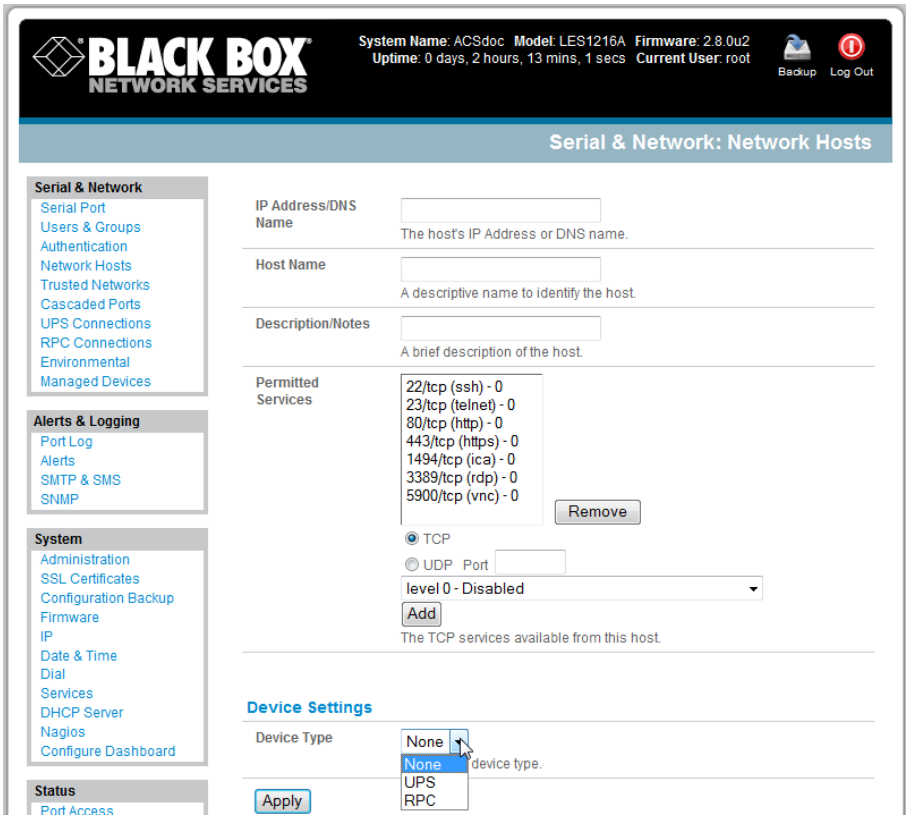
4.3 Authentication

Refer to *Chapter 9.1— Remote Authentication Configuration* for authentication configuration details.

4.4 Network Hosts

To access a locally networked computer or device (referred to as a *Host*), you must identify the Host and specify the TCP or UDP ports/services that will be used to control that Host.

- Selecting **Serial & Network: Network Hosts** presents all the network connected Hosts that have been enabled for access, and the related access TCP ports/services.
- Click **Add Host** to enable access to a new Host (or select **Edit** to update the settings for an existing Host).

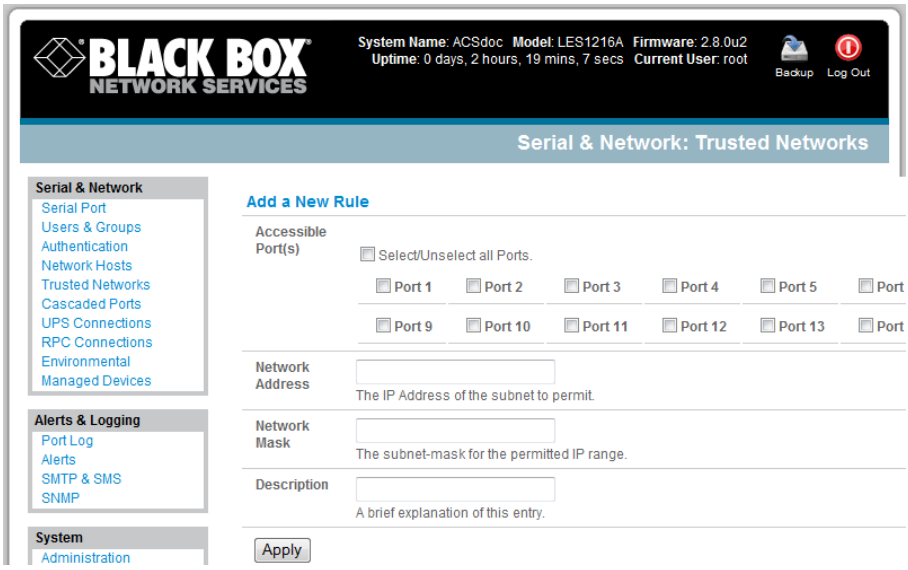


- Enter the **IP Address** or **DNS Name** and a **Host Name** (up to 254 alphanumeric characters) for the new network connected Host (and optionally enter a **Description**).
- Add or edit the **Permitted Services** (or TCP/UDP port numbers) that are authorized to be used in controlling this host. Only these *permitted services* will be forwarded through by SDT to the Host. All other services (TCP/UDP ports) will be blocked.
- The **Logging Level** specifies the level of information to be logged and monitored for each Host access (refer to *Chapter 7—Alerts and Logging*).
- If the Host is a PDU or UPS power device or a server with IPMI power control, then specify **RPC** (for IPMI and PDU) or **UPS** and the **Device Type**. The *Administrator* can then configure these devices and enable which users have permission to remotely cycle power, etc. (refer to *Chapter 8*). Otherwise, leave the Device Type set to None.
- If the *console server* has been configured with distributed Nagios monitoring enabled, then you will also be presented with **Nagios Settings** options to enable nominated services on the Host to be monitored (refer to *Chapter 10— Nagios Integration*).
- Click **Apply.** This will create the new Host and also create a new Managed Device (with the same name).

4.5 Trusted Networks

The **Trusted Networks** facility gives you an option to nominate specific IP addresses where users (*Administrators* and *Users*) must be located to access *console server* serial ports.

- Select **Serial & Network: Trusted Networks.**
- To add a new trusted network, select **Add Rule**.



- Select the **Accessible Port(s)** that the new rule is to be applied to.
- Then, enter the **Network Address** of the subnet to be permitted access.
- Then, specify the range of addresses that are to be permitted by entering a **Network Mask** for that permitted IP range, *for example:*
  - To permit all the users located with a particular Class C network (for example, 204.15.5.0) connection to the nominated port then you would add the following Trusted Network New Rule:

Network Address	204.15.5.0
Network Mask	255.255.255.0

  - If you want to permit only the one user who is located at a specific IP address (for example, 204.15.5.13 say) to connect:

Network Address	204.15.5.0
Network Mask	255.255.255.255

  - If, however, you want to allow all the users operating from within a specific range of IP addresses (for example, any of the thirty addresses from 204.15.5.129 to 204.15.5.158) to be permitted connection to the nominated port:

Host /Subnet Address	204.15.5.128
Subnet Mask	255.255.255.224

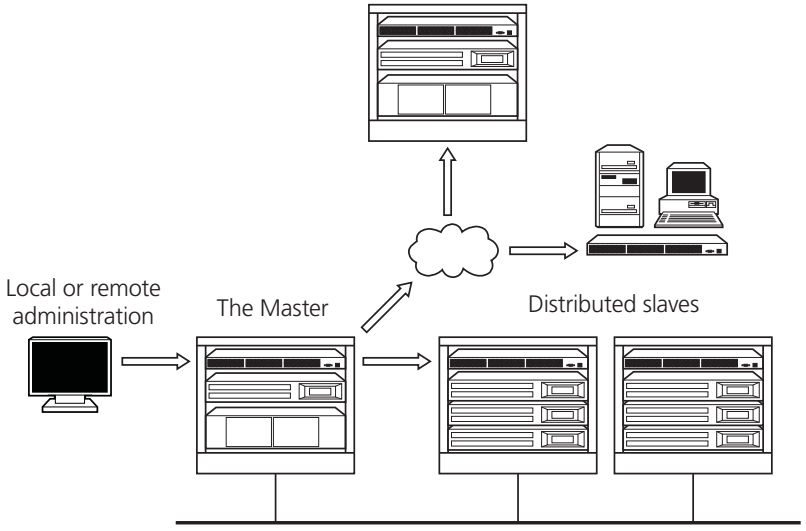
- Click **Apply.**

Note The above Trusted Networks will limit *Users* and *Administrators* access to the console serial ports. They do not restrict access to the *console server* itself or to attached hosts. To change the default settings for this access, you will to need to edit the *IPtables* rules as described in *Chapter 14—Advanced.*

4.6 Serial Port Cascading

Cascaded Ports enables you to cluster distributed *console server*s. A large number of serial ports (up to 1000) can be configured and accessed through one IP address and managed through one Management Console. One *console server*, the Master, controls other *console server*s as Slave units and all the serial ports on the Slave units appear as if they are part of the Master.

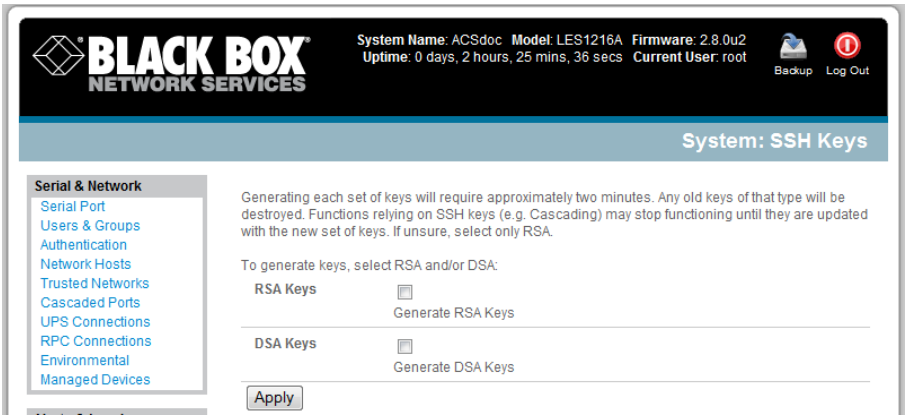
Black Box's clustering connects each Slave to the Master with an SSH connection. This uses public key authentication so the Master can access each Slave using the SSH key pair (rather than using passwords). This ensures secure authenticated communications between Master and Slaves, enabling the Slave *console server* units to be distributed locally on a LAN or remotely around the world.



4.6.1 Automatically generate and upload SSH keys

To set up public key authentication, you must first generate an RSA or DSA key pair and upload them into the Master and Slave *console server*s. This can all be done automatically from the Master.

- Select **System: Administration** on Master's Management Console.
- Check **Generate SSH keys automatically** and click **Apply.**


BLACK BOX
NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 2 hours, 25 mins, 36 secs Current User: root
Backup
Log Out
System: SSH Keys
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Environmental
Managed Devices
Generating each set of keys will require approximately two minutes. Any old keys of that type will be destroyed. Functions relying on SSH keys (e.g. Cascading) may stop functioning until they are updated with the new set of keys. If unsure, select only RSA.
To generate keys, select RSA and/or DSA:
RSA Keys
Generate RSA Keys
DSA Keys
Generate DSA Keys
Apply

Next, you must select whether to generate keys using RSA and/or DSA (if unsure, select only RSA). Generating each set of keys will require approximately two minutes, and the new keys will destroy any old keys of that type that may previously been uploaded.

Also, while the new generation is underway on the master, functions relying on SSH keys (for example, cascading) may stop functioning until they are updated with the new set of keys.

To generate keys:

- Select **RSA Keys** and/or **DSA Keys.**
- Click **Apply.**
- Once the new keys have been successfully generated, **Click here to return** and the keys will automatically be uploaded to the Master and connected Slaves.

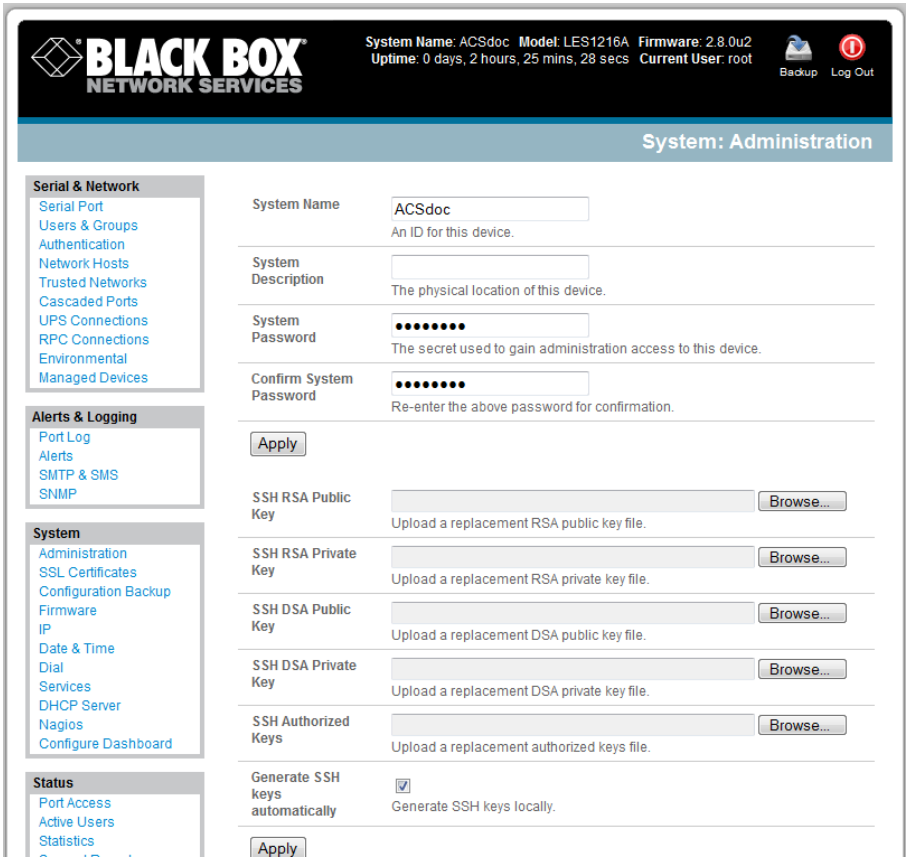
4.6.2 Manually generate and upload SSH keys

Or, if you have an RSA or DSA key pair. you can manually upload them to the Master and Slave *console server*s.

Note If you already have an RSA or DSA key pair that you do not want to use, you will need to create a key pair using *ssh-keygen, PuTTYgen* or a similar tool as detailed in Chapter 15.6.

To manually upload the public and private key pair to the Master *console server*:

- Select **System: Administration** on Master's Management Console.
- Browse to the location where you have stored RSA (or DSA) Public Key and upload it to **SSH RSA (DSA) Public Key**.
- Browse to the stored RSA (or DSA) Private Key and upload it to **SSH RSA (DSA) Private Key.**
- Click **Apply.**



Next, you must register the Public Key as an Authorized Key on the Slave. In a case that has only one Master with multiple Slaves, you only need to upload the one RSA or DSA public key for each Slave.

Note Using key pairs can be confusing since one file (Public Key) fulfills two roles— Public Key and Authorized Key. For a more detailed explanation, refer to the *Authorized Keys* section of Chapter 15.6. Also, refer to this chapter if you need to use more than one set of Authorized Keys in the Slave.

- Select **System: Administration** on the Slave's Management Console.
- Browse again to the stored RSA (or DSA) Public Key and upload it to Slave's **SSH Authorized Key.**
- Click **Apply.**

The next step is to *Fingerprint* each new Slave-Master connection. This one-time step will validate that you are establishing an SSH session to who you think you are. On the first connection, the Slave will receive a *fingerprint* from the Master which will be used on all future connections:

- To establish the fingerprint, first log in the Master server as *root* and establish an SSH connection to the Slave remote host:

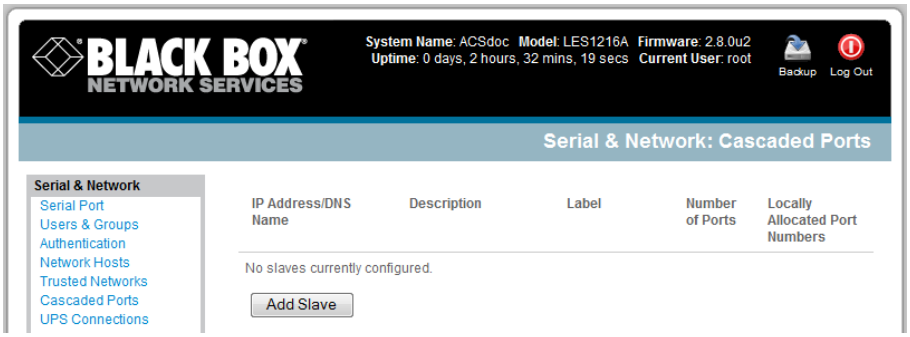
  # ssh remhost

Once the SSH connection has been established, the system asks you to accept the key. Answer *yes* and the *fingerprint* will be added to the list of known hosts. For more details on Fingerprinting, refer to Chapter 15.6.

- If the system asks you to supply a password, then there is a problem with uploading keys. The keys should remove any need to supply a password.

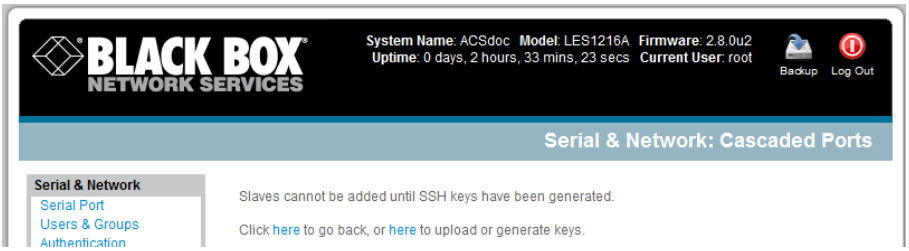
4.6.3 Configure the slaves and their serial ports

You can now begin setting up the Slaves and configuring Slave serial ports from the Master *console server*:



- Select **Serial & Network: Cascaded Ports** on the Master's Management Console:
- To add clustering support, select **Add Slave**.

Note You can't add any Slaves until you automatically or manually generate SSH keys.


BLACK BOX
NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 2 hours, 33 mins, 23 secs Current User: root
Backup
Log Out
Serial & Network: Cascaded Ports
Serial & Network
Serial Port
Users & Groups
Authentication
Slaves cannot be added until SSH keys have been generated.
Click here to go back, or here to upload or generate keys.

To define and configure a Slave:

- Enter the remote **IP Address** (or DNS Name) for the Slave *console server*.
- Enter a brief **Description** and a short **Label** for the Slave (use a convention here that enables you to effectively manage large networks of clustered *console server*s and the connected devices).
- Enter the full number of serial ports on the Slave unit in **Number of Ports**.
- Click **Apply**. This will establish the SSH tunnel between the Master and the new Slave.

The **Serial & Network: Cascaded Ports** menu displays all the Slaves and the port numbers that have been allocated on the Master. If the Master *console server* has 16 ports of its own, then ports 1-16 are pre-allocated to the Master. The first Slave added will be assigned port number 17 and up.

Once you have added all the Slave *console server*s, you can assign and access the Slave serial ports and the connected devices from the Master's Management Console menu. You can also access them through the Master's IP address.

- Select the appropriate **Serial & Network: Serial Port** and **Edit** to configure the serial ports on the Slave.

- Select the appropriate **Serial & Network: Users & Groups** to add new users with access privileges to the Slave serial ports (or to extend existing users' access privileges).
- Select the appropriate **Serial & Network: Trusted Networks** to specify network addresses that can access nominated Slave serial ports .
- Select the appropriate **Alerts & Logging: Alerts** to configure Slave port Connection, State Change, or Pattern Match alerts.
- The configuration changes made on the Master are propagated out to all the Slaves when you click **Apply**.

4.6.4 Managing the Slaves

The Master is in control of the Slave serial ports. For example, if you change *User* access privileges or edit any serial port setting on the Master, the updated configuration files will be sent out to each Slave in parallel. Each Slave will then automatically make changes to its local configuration (and only make those changes that relate to its particular serial ports).

You can still use the local Slave Management Console to change the settings on any Slave serial port (such as alter the baud rates). These changes will be overwritten next time the Master sends out a configuration file update.

Also, while the Master is in control of all Slave serial port related functions, it is not master over the Slave network host connections or over the Slave *console server* system itself.

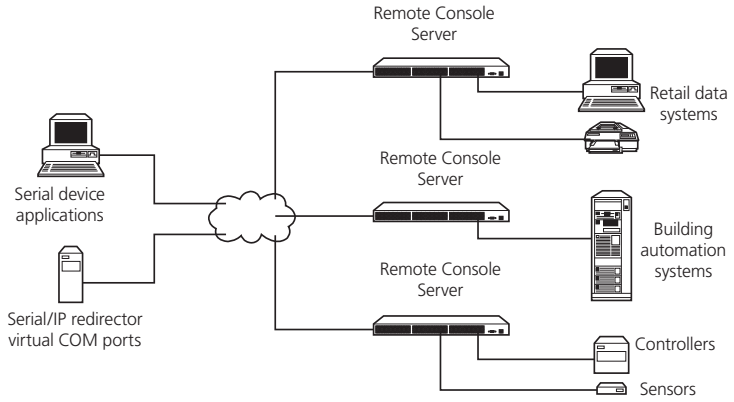
You must access each Slave directly to manage Slave functions such as IP, SMTP & SNMP Settings, Date &Time, and DHCP server. These functions are not overwritten when configuration changes are propagated from the Master. Similarly, you have to configure the Slaves Network Host and IPMI settings at each Slave.

The Master's Management Console provides a consolidated view of the settings for its own and all the Slave's serial ports. The Master does not provide a fully consolidated view. For example, if you want to find out who's logged in to cascaded serial ports from the master, you'll see that *Status: Active Users* only displays those users active on the Master's ports, so you may need to write custom scripts to provide this view. This is covered in Chapter 11.

4.7 Serial Port Redirection

To allow an application on a client PC to access the virtual serial ports on the console server, you need to run client software (to redirect the local serial port traffic to remote *console server* serial port).

There's a selection of commercial software available including *Serial to Ethernet* from Eltima (www.eltima.com) and *Serial/IP™ COM Port Redirector* from Tactical Software (www.tacticalsoftware.com/products/serialip.htm).

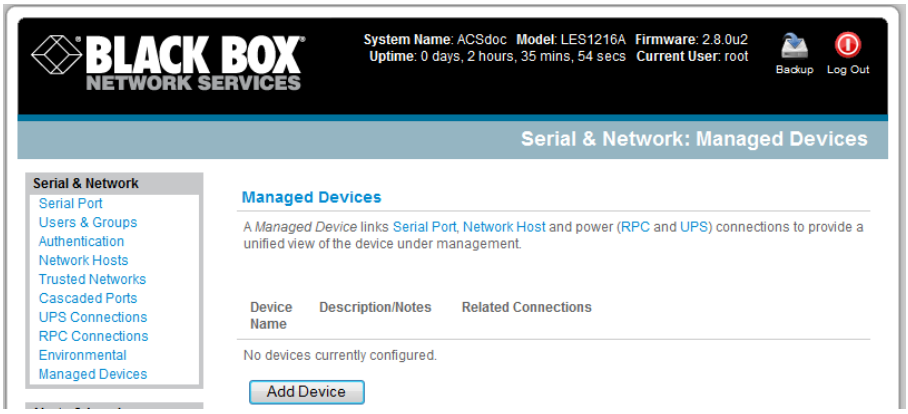


This serial port redirector software is loaded in your desktop PC, and it allows you to use a serial device that's connected to the remote *console server* as if it were connected to your local serial port.

4.8 Managed Devices

Managed Devices presents a consolidated view of all the connections to a device that you can access and monitor through the *console server*. To view the connections to the devices:

- Select **Serial & Network: Managed Devices**.



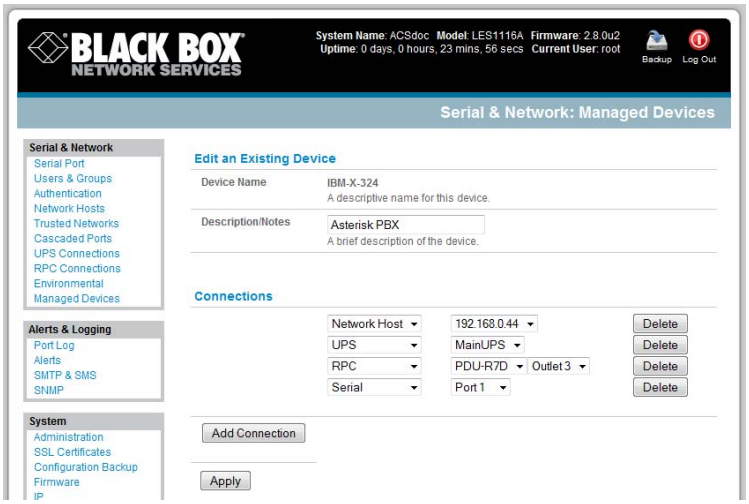
This screen displays all the Managed Devices with their Description/Notes. It also lists all the configured Connections, that is, Serial Port # (if serially connected) or USB if USB connected; IP Address (if network connected); Power PDU/outlet details (if applicable), and any UPS connections. Devices such as servers will commonly have more than one power connections (for example, dual power supplied) and more than one network connection (for example, for BMC/service processor).

All *Users* can view (but not edit) these Managed Device connections by selecting Manage: Devices. The *Administrator* user can edit and add/delete these Managed Devices and their connections.

To edit an existing device and add a new connection:

- Select **Edit** on the **Serial & Network: Managed Device**s and click **Add Connection**.

- Select the connection type for the new connection (Serial, Network Host, UPS, or RPC) and then select the specific connection from the presented list of configured unallocated hosts/ports/outlets.



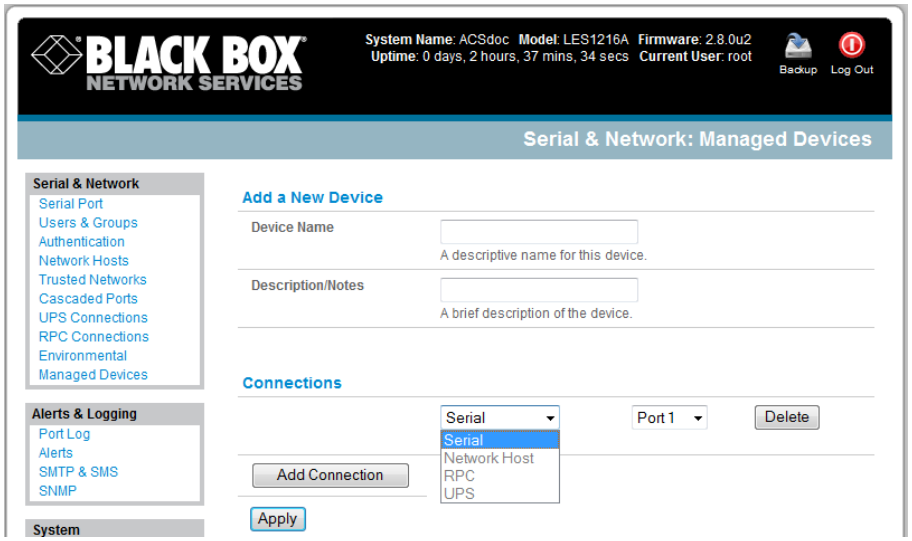
To add a new network-connected Managed Device:

- The *Administrator* adds a new network-connected Managed Device using **Add Host** on the **Serial & Network: Network Host** menu. This automatically creates a corresponding new Managed Device (as covered in *Section 4.4—Network Hosts).*
- When adding a new network-connected RPC or UPS power device, you set up a Network Host, designate it as RPC or UPS, then go to **RPC Connections** (or **UPS Connections**) to configure the relevant connection. A corresponding new Managed Device (with the same Name /Description as the RPC/UPS Host) is not created until you complete this connection step (refer *Chapter 8—Power and Environment).*

Note The outlet names on this newly created PDU will by default be "Outlet 1" and "Outlet 2." When you connect a particular Managed Device (that draws power from the outlet), then the outlet will take the powered Managed Device's name.

To add a new serially connected Managed Device:

- Configure the serial port using the **Serial & Network: Serial Port** menu (refer to *Section 4.1—Configure Serial Port).*
- Select **Serial & Network: Managed Devices** and click **Add Device**.
- Enter a **Device Name** and **Description** for the Managed Device.



- Click **Add Connection** and select **Serial** and the **Port** that connects to the Managed Device.
- To add a UPS/RPC power connection or network connection or another serial connection, click **Add Connection.**
- Click **Apply.**

Note To set up a new serially connected RPC UPS or EMD device, configure the serial port, designate it as a Device, then enter a Name and Description for that device in the **Serial & Network: RPC Connections** (or **UPS Connections** or **Environmental**). When applied, this will automatically create a corresponding new Managed Device with the same Name /Description as the RPC/UPS Host (refer to *Chapter 8—Power and Environment)*.

All the outlet names on the PDU will by default be “Outlet 1” and “Outlet 2.” When you connect a particular Managed Device (that draws power from the outlet) then the outlet will then take up the name of the powered Managed Device.

Chapter 5 Failover and OoB Dial Access

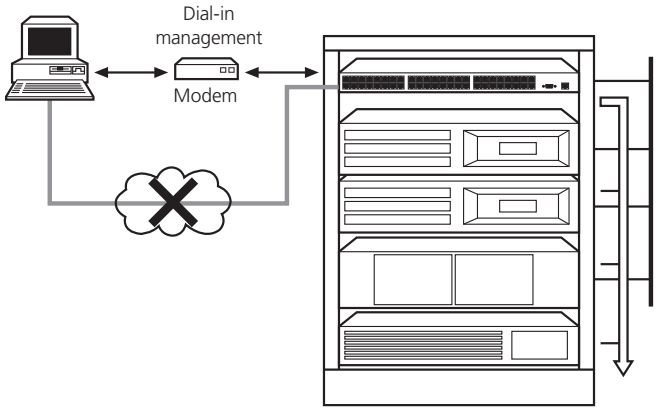
Introduction

The *console server* has a number of fail-over and out-of-band access capabilities to make sure it's available if there are difficulties accessing the console server through the principal network path. This chapter covers:

- out-of-band (OoB) access from a remote location using dial-up modem.
- out-dial failover.
- OoB access using an alternate broadband link (LES1208A, LES1216A, and LES1248A models only).
- broadband failover.

5.1 OoB Dial-In Access

To enable OoB dial-in access, you first configure the *console server.* Once it's set up for dial-in PPP access, the *console server* will await an incoming dial-in connection. Set up the remote client dial-in software so it can establish a network connection from the *Administrator's* client modem to the dial-in modem on the *console server*.



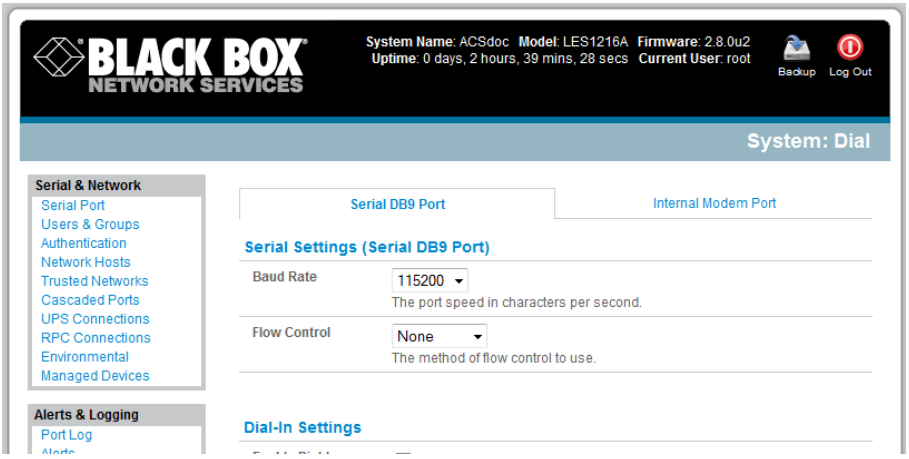
Note The LES1208A, LES1216A, and LES1248A models all have an internal modem and a DB9 Local/Console port for OoB access. With these models, you can still attach an external modem via a serial cable to the DB9 port, and you can configure the second Ethernet port for broadband OoB access.

Make sure you unplug the *console server* power before installing the modem. When it next boots, it will detect the modem and a *PC Card Modem* tab will appear under *System -> Dial.*

The LES1108A, LES1116A, and LES1148A models need to have an external modem attached via a serial cable to the DB9 port marked *Local* (located on the front of the unit).

5.1.1 Configure Dial-In PPP

To enable dial-in PPP access on the modem:



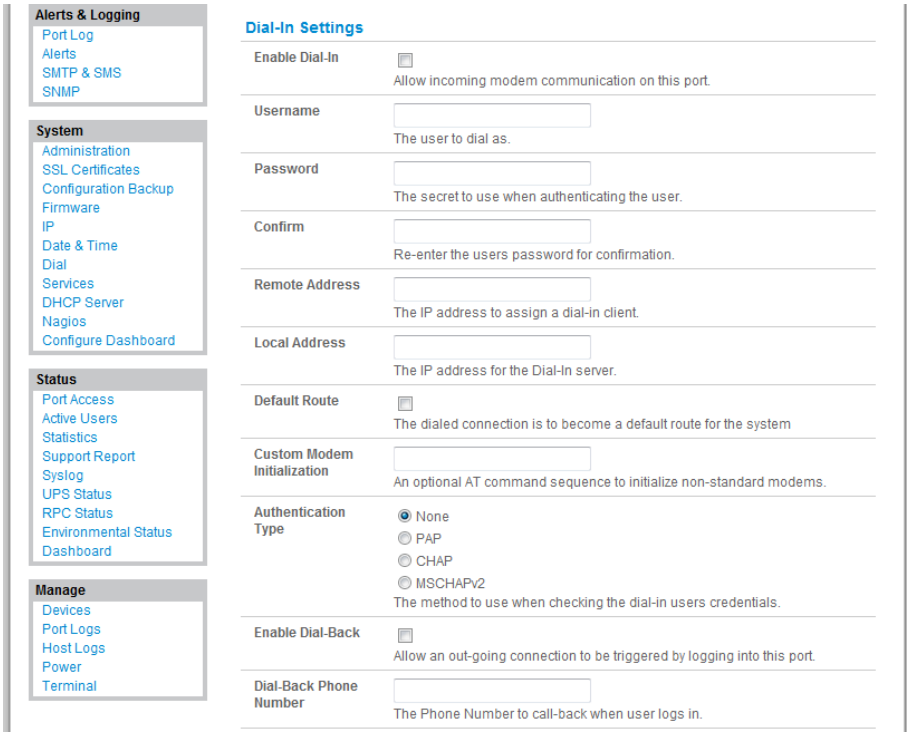
- Select the **System: Dial** menu option and the port to be configured (**Serial DB9 Port** or **Internal Modem Port**).

Note The *console server* console/modem serial port is set by default to 115200 baud, No parity, 8 data bits and 1 stop bit, with software (Xon-Xoff) flow control enabled for the Serial DB9 Port and 9600 baud for the Internal modem and PC Card Ports. When enabling OoB dial-in, we recommend that this be changed to 38,4000 baud with Hardware Flow Control.

- Select the **Baud Rate** and **Flow Control** that will communicate with the modem.

Note You can further configure the console/modem port (*for example*, to include *modem init* strings) by editing */etc/mgetty.config* files as described in the *Chapter 15—Advanced Configuration.*

- Check the **Enable Dial-In Access** box.
- Enter the **User name** and **Password** to be used for the dial-in PPP link.
- In the **Remote Address** field, enter the IP address to be assigned to the dial-in client. You can select any address for the Remote IP Address. It, and the Local IP Address, must both be in the same network range (*e.g.* 200.100.1.12 and 200.100.1.67).
- In the **Local Address** field, enter the IP address for the Dial-In PPP Server. This is the IP address that will be used by the remote client to access *console server* once the modem connection is established. You can select any address for the Local IP Address but it must be in the same network range as the Remote IP Address.
- The **Default Route** option enables the dialed PPP connection to become the default route for the *Console server*.
- The **Custom Modem Initialization** option allows you to enter a custom AT string modem initialization string (*for example,* AT&C1&D3&K3).



- You must select the **Authentication Type** to apply to the dial-in connection. The *console server* uses authentication to challenge *Administrators* who dial-in to the *console server*. (For dial-in access, the username and password received from the dial-in client are verified against the local authentication database stored on the *console server*). The *Administrator* must also configure the client PC/workstation to use the selected authentication scheme. Select **PAP, CHAP, MSCHAPv2,** or **None,** and click **Apply.**

  None With this selection, no username or password authentication is required for dial-in access. We do not recommend this.

  PAP Password Authentication Protocol (PAP) is the usual method of user authentication used on the internet: sending a username and password to a server where they are compared with a table of authorized users. While most common, PAP is the least secure of the authentication options.

  CHAP Challenge-Handshake Authentication Protocol (CHAP) is used to verify a user's name and password for PPP Internet connections. It is more secure than PAP, the other main authentication protocol.

  MSCHAPv2 Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is authentication for PPP connections between a computer using a Microsoft Windows operating system and a network access server. It is more secure than PAP or CHAP, and is the only option that also supports data encryption.

- *Console servers* support dial-back for additional security. Check the **Enable Dial-Back** box and enter the phone number to call to re-establish an OoB link once a dial-in connection is logged.

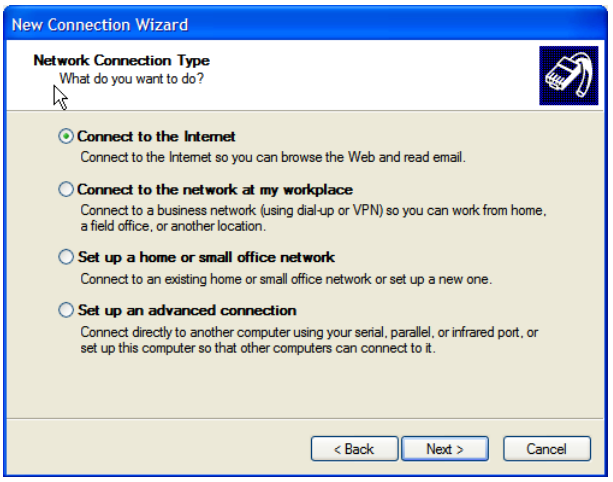
Note *Chapter 15—Advanced Configuration*) has examples of Linux commands that you can use to control the modem port operation at the command line level.

5.1.2 Using SDT Connector client

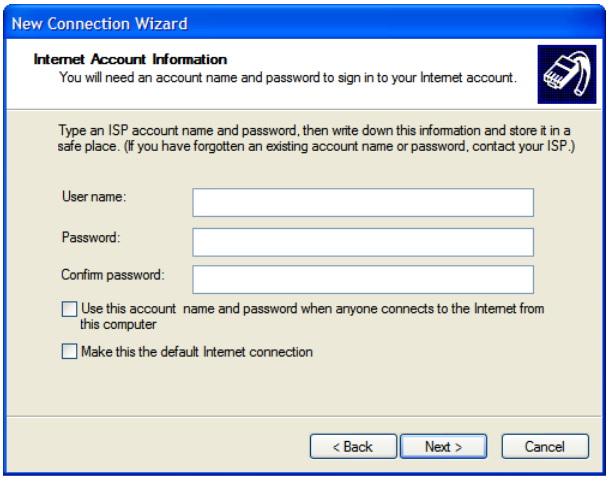
*Administrator*s can use their *SDT Connector* client to set up secure OoB dial-in access to all their remote console servers. With a point and click, you can initiate a dial up connection. Refer to *Chapter 6.5.*

5.1.3 Set up Windows XP/ 2003/Vista/7 client

- Open **Network Connections** in Control Panel and click the **New Connection Wizard.**


New Connection Wizard
Network Connection Type
What do you want to do?
Connect to the Internet
Connect to the Internet so you can browse the Web and read email.
Connect to the network at my workplace
Connect to a business network (using dial-up or VPN) so you can work from home, a field office, or another location.
Set up a home or small office network
Connect to an existing home or small office network or set up a new one.
Set up an advanced connection
Connect directly to another computer using your serial, parallel, or infrared port, or set up this computer so that other computers can connect to it.
< Back
Next >
Cancel

- Select **Connect to the Internet** and click **Next.**
- On the **Getting Ready** screen, select **Set up my connection manually** and click **Next.**
- On the **Internet Connection** screen, select **Connect using a dial-up modem** and click **Next.**
- Enter a **Connection Name** (any name you choose) and the dial-up **Phone number** that will connect through to the *console server* modem.


New Connection Wizard
Internet Account Information
You will need an account name and password to sign in to your Internet account.
Type an ISP account name and password, then write down this information and store it in a safe place. (If you have forgotten an existing account name or password, contact your ISP.)
User name:
Password:
Confirm password:
Use this account name and password when anyone connects to the Internet from this computer
Make this the default Internet connection
< Back
Next >
Cancel

- Enter the PPP **User name** and **Password** you set up for the *console server.*

5.1.4 Set up earlier Windows clients

- For Windows 2000, the PPP client set up procedure is the same as above, except you get to the **Dial-Up Networking Folder** by clicking the **Start** button and selecting **Settings.** Then, click **Network and Dial-up Connections** and click **Make New Connection.**
- Similarly, for Windows 98, you double click **My Computer** on the Desktop, then open **Dial-Up Networking** and double click **Make New Connection**. Then, proceed as above.

5.1.5 Set up Linux clients for dial-in

The online tutorial http://www.yolinux.com/TUTORIALS/LinuxTutorialPPP.html presents a selection of methods for establishing a dial up PPP connection:

- Command line PPP and manual configuration (works with any Linux distribution).
- Using the *Linuxconf* configuration tool (for Red Hat compatible distributions). This configures the scripts *ifup/ifdown* to start and stop a PPP connection.
- Using the Gnome control panel configuration tool.
- WVDIAL and the Redhat "Dialup configuration tool" .
- GUI dial program X-isp. Download/Installation/Configuration.

Note For all PPP clients:

- Set the PPP link up with TCP/IP as the only protocol enabled.
- Specify that the Server will assign IP address and do DNS.
- Do not set up the console server PPP link as the default for Internet connection.

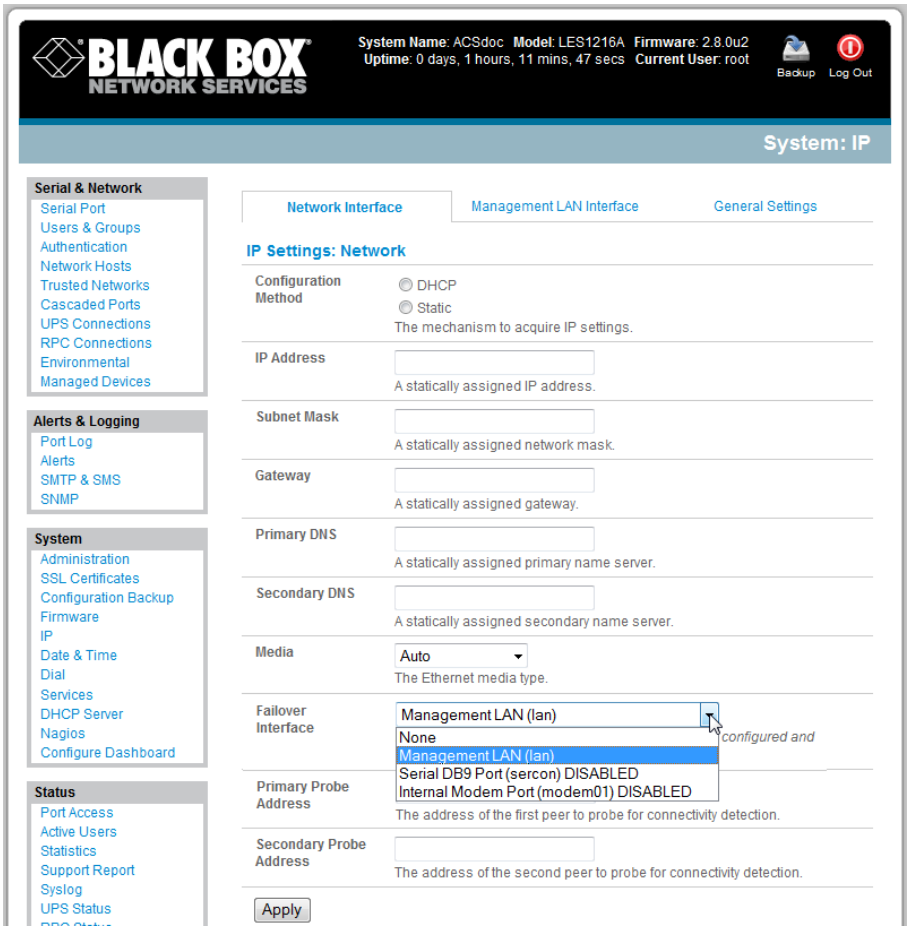
5.2 OoB broadband access

The LES1208A, LES1216A, and LES1248A *console server*s have a second Ethernet port (*Network 2*) that you can configure for alternate and OoB (out-of-band) broadband access. With two active broadband access paths to the *console server*, if you are unable to access it through the primary management network (*Network or Network1*), you can still access it through the alternate broadband path (for example, a T1 link).

- On the **System: IP** menu select **Network 2** and configure the **IP Address, Subnet Mask**, **Gateway,** and **DNS** with the access settings for the alternate link.
- Make sure that when you configure the principal **Network 1 Settings** connection, the **Failover Interface** is set to *None.*

5.3 Broadband Ethernet Failover

The second Ethernet port on the LES1208A, LES1216A, and LES1248A Advanced Console Servers can also be configured for failover to ensure transparent high availability.



- When configuring the principal network connection, specify **Network 2 (eth1)** as the **Failover Interface** to use when a fault is detected with Network 1 (eth0).
- Specify the **Probe Addresses** of two sites (the **Primary** and **Secondary**) that the Advanced Console Server is to *ping* to determine if Network 1 (eth0) is still operating.
- On the **Management LAN Interface** - **Network 2,** configure the **IP Address/Subnet Mask/Gateway** the same as **Network Interface** - **Network 1**.

In this mode, Network 2 (eth1) is available as the transparent back-up port to Network 1 (eth0) for accessing the management network. Network 2 will automatically and transparently take over the work of Network 1, if Network 1 becomes unavailable for any reason. When Network 1 becomes available again, it takes over the work again.

5.4 Dial-Out Failover

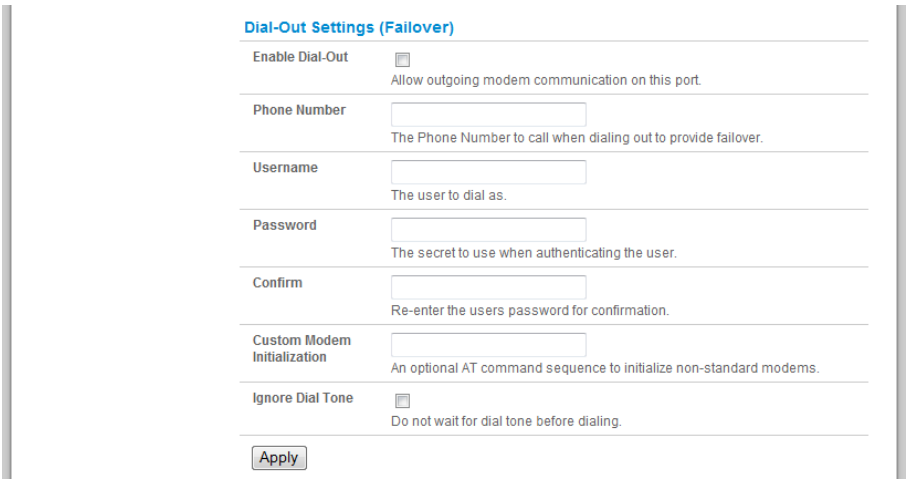
The *console server*s can be configured so a dial-out PPP connection is automatically set up in case the principal management network is disrupted:

- When configuring the principal network connection in **System: IP**, specify **Internal Modem** (or the **Dial Serial DB9** if you are using an external modem on the Console port) as the **Failover Interface** to use when a fault is detected with Network1 (eth0).

- Specify the **Probe Addresses** of two sites (the **Primary** and **Secondary**) that the console server is to *ping* to determine if Network1 is still operating.
- Select the **System: Dial** menu option and the port to be configured (**Serial DB9 Port** or **Internal Modem Port**).
- Select the **Baud Rate** and **Flow Control** that will communicate with the modem.

Note You can further configure the console/modem port (*for example,* to include *modem init* strings) by editing */etc/mgetty.config* files as described in Chapter 13.

- Check the **Enable Dial-Out** box in **System: Dial** and enter the access details to call the remote PPP server.

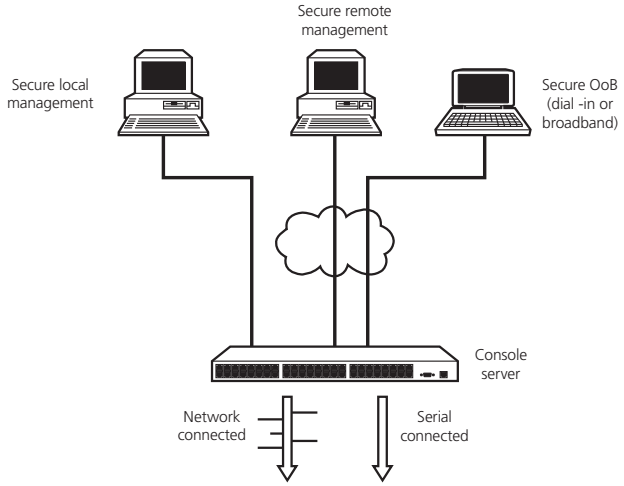

Dial-Out Settings (Failover)
Enable Dial-Out
Allow outgoing modem communication on this port.
Phone Number
The Phone Number to call when dialing out to provide failover.
Username
The user to dial as.
Password
The secret to use when authenticating the user.
Confirm
Re-enter the users password for confirmation.
Custom Modem Initialization
An optional AT command sequence to initialize non-standard modems.
Ignore Dial Tone
Do not wait for dial tone before dialing.
Apply

Chapter 6 Secure SSH Tunneling & SDT Connector

Introduction

Each Black Box *console server* has an embedded SSH server and uses SSH tunneling so remote users can securely connect through the *console server* to Managed Devices—using text-based console tools (such as SSH, telnet, SoL) or graphical tools (such VNC, RDP, HTTPS, HTTP, X11, VMware, DRAC, iLO).

The Managed Devices you access can be located on the same local network as the *console server* or they can be attached to the *console server* via a serial port. The remote *User/Administrator* connects to the *console server* thru an SSH tunnel via dial-up, wireless or ISDN modem; a broadband Internet connection; the enterprise VPN network; or the local network.



To set up the secure SSH tunnel from the client PC to the *console server*, install and launch SSH client software on the User/*Administrator*'s PC. Black Box recommends you use the *SDT Connector* client software supplied with the *console server* for this. *SDT Connector* is simple to install and auto-configure and it provides all your users with point-and-click access to all the systems and devices in the secure network. With one click, *SDT Connector* sets up a secure SSH tunnel from the client to the selected *console server*, then establishes a port forward connection to the target network connected host or serial connected device. Next, it executes the client application that it uses in communicating with the host.

This chapter details the basic SDT Connector operations:

- Configuring the *console server* for SSH tunneled access to network attached hosts and setting up permitted Services and user access (*Section 6.1*).
- Setting up the SDT Connector client with gateway, host, service, and client application details, and making connections between the Client PC and hosts connected to the *console server* (*Section 6.2*).
- Using SDT Connector to access the Management Console via a browser (*Section 6.3*).

- Using SDT Connector to Telnet or SSH connect to devices that are serially attached to the *console server* (*Section 6.4*).

The chapter then covers more advanced SDT Connector and SSH tunneling topics:

- Using SDT Connector for out-of-band access (*Section 6.5*).
- Automatic importing and exporting configurations (*Section 6.6*).
- Configuring Public Key Authentication (*Section 6.7*).
- Setting up a SDT Secure Tunnel for Remote Desktop (*Section 6.8*).
- Setting up a SDT Secure Tunnel for VNC (*Section 6.9*).
- Using SDT to IP connect to hosts that are serially attached to the *console server* (*Section 6.10*).

6.1 Configuring for SSH Tunneling to Hosts

To set up the *console server* to SSH tunnel access a network attached *host*:

➢ Add the new *host* and the *permitted services* using the **Serial & Network: Network Hosts** menu as detailed in *Network Hosts (Chapter 4.4).* Only these *permitted services* will be forwarded through by SSH to the *host*. All other services (TCP/UDP ports) will be blocked.

Note Following are some of the TCP Ports used by SDT in the *console server*:

22	SSH (All SDT Tunneled connections)
23	Telnet on local LAN (forwarded inside tunnel)
80	HTTP on local LAN (forwarded inside tunnel)
3389	RDP on local LAN (forwarded inside tunnel)
5900	VNC on local LAN (forwarded inside tunnel)
73XX	RDP over serial from local LAN – where XX is the serial port number (that is, 7301 to 7348 on a 48 port *console server*)
79XX	VNC over serial from local LAN – where XX is the serial port number

➢ Add the new *Users* using **Serial & Network: Users & Groups** menu as detailed in *Network Hosts (Chapter 4.4). Users* can be authorized to access the *console server* ports and specified network attached hosts. To simplify configuration, the *Administrator* can first set up *Groups* with group access permissions, then *Users* can be classified as members of particular *Groups*.

6.2 SDT Connector Client Configuration

The *SDT Connector* client works with all Black Box *console server*s. Each of these remote *console server*s has an embedded OpenSSH based server that you can configure to *port forward* connections from the *SDT Connector* client to hosts on their local network (as detailed in the previous chapter). You can also pre-configure the *SDT Connector* with the access tools and applications that are available to run when you've established access to a particular host.

SDT Connector can connect to the *console server* using an alternate OoB access. It can also access the *console server* itself and access devices connected to serial ports on the *console server*.

6.2.1 SDT Connector installation

- The *SDT Connector* set up program (***SDTConnector Setup-1.n.exe*** or ***sdtcon-1.n.tar.gz***) is included on the CD supplied with your Black Box *console server*.
- Run the set-up program.

SDTConnector Setup
Welcome to the
SDTConnector Setup Wizard
This wizard will guide you through the installation of SDTConnector.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel

Note For Windows clients, the *SDTConnectorSetup-1.n.exe* application will install the *SDT Connector 1.n.exe* and the config file *defaults.xml.* If there is already a config file on the Windows PC, then it will not be overwritten. To remove an earlier config file, run the *regedit* command and search for "SDT Connector," then remove the directory with this name.

For Linux and other Unix clients, *SDTConnector.tar.gz* application will install the *sdtcon-1.n.jar* and the config file *defaults.xml.*

Once the installer completes you will have a working *SDT Connector* client installed on your machine and an icon on your desktop:

- Click the *SDT Connector* icon on your desktop to start the client.

Note *SDT Connector* is a Java application, so it must have a Java Runtime Environment (JRE) installed. You can download this for free from http://java.sun.com/j2se/. It installs on Windows 2000, XP, 2003, Vista, and 7 PCs and on most Linux platforms. Solaris platforms are also supported, but they must have Firefox installed. *SDT Connector* can run on any system with Java 1.4.2 and above installed, but it assumes the web browser is Firefox, and that *xterm -e telnet* opens a telnet window.

To operate *SDT Connector,* you first need to add new gateways to the client software by entering the access details for each *console server* (refer to *Section 6.2.2*). Then, let the client auto-configure all host and serial port connections from each *console server* (refer to *Section 6.2.3*). Finally, point-and-click to connect to the Hosts and serial devices (refer to *Section 6.2.4*).

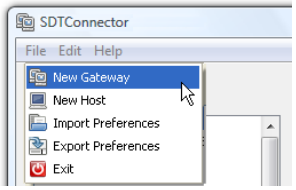
Or, you can manually add network connected hosts (refer to *Section 6.2.5*) and manually configure new services to use to access the *console server* and the hosts (refer to *Section 6.2.6*). Then, manually configure clients to run on the PC that will use the service to connect to the hosts and serial port devices

(refer to *Section 6.2.7 and 6.2.9*). You can also set up *SDT Connector* to connect out-of-band to the *console server* (refer to *Section 6.2.9*).

6.2.2 Configuring a new console server gateway in the SDT Connector client

To create a secure SSH tunnel to a new *console server*:

- Click the *New Gateway* icon or select the **File: New Gateway** menu option.

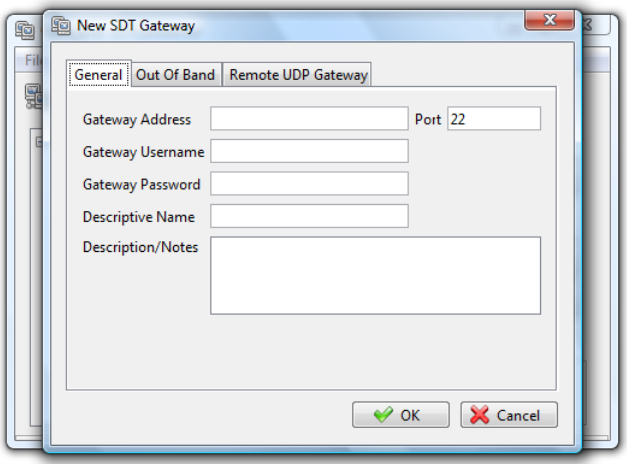


- Enter the IP or DNS **Address** of the *console server* and the SSH port that you will use (typically 22).

Note If *SDT Connector* is connecting to a remote *console server* through the public Internet or routed network you will need to:

- Determine *the public IP address* of the *console server* (or of the router/ firewall that connects the *console server* to the Internet) as assigned by the ISP. One way to find the public IP address is to access http://checkip.dyndns.org/ or http://www.whatismyip.com/ from a computer on the same network as the *console server* and note the reported IP address.
- Set port forwarding for TCP port 22 through any firewall/NAT/router that is located between *SDT Connector* and the *console server* so it points to the *console server*. http://www.portforward.com has port forwarding instructions for a range of routers. Also, you can use the Open Port Check tool from http://www.canyouseeme.org to check if port forwarding through local firewall/NAT/router devices has been properly configured.

- Enter the **Username** and **Password** of a user on the gateway that is enabled to connect via SSH and/or create SSH port redirections.

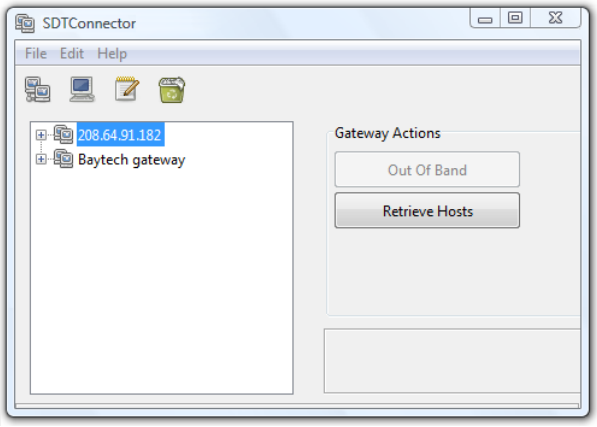


- Or, enter a **Descriptive Name** to display instead of the IP or DNS address, and any **Notes** or a **Description** of this gateway (such as its firmware version, site location, or anything special about its network configuration).
- Click **OK** and an icon for the new gateway will now appear in the *SDT Connector* home page.

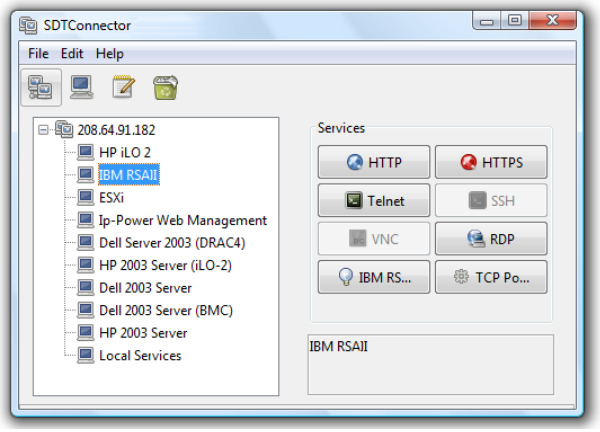
Note For an *SDT Connector* user to access a *console server* (and then access specific hosts or serial devices connected to that *console server*), that user must first be setup on the *console server*, and must be authorized to access the specific ports/hosts (refer to Chapter 5). Only these *permitted services* will be forwarded through by SSH to the Host. All other services (TCP/UDP ports) will be blocked.

6.2.3 Auto-configure SDT Connector client with the user's access privileges

Each user on the *console server* has an access profile that was configured with those specific connected hosts and serial port devices the user has authority to access, and a specific set of the enabled services for each of these. You can upload this configuration automatically into the SDT Connector client:



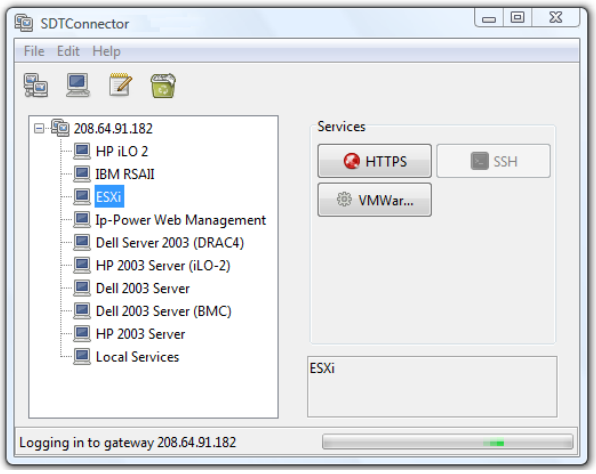
- Click on the new gateway icon and select **Retrieve Hosts**. This will:
  - configure access to network connected Hosts that the user is authorized to access and set up (for each of these Hosts) the services (for example, HTTPS, IPMI2.0) and the related IP ports being redirected.
  - configure access to the *console server* itself (this is shown as a *Local Services* host).
  - configure access with the enabled services for the serial port devices connected to the *console server.*



Note The Retrieve Hosts function will auto-configure all user classes (that is, they can be members of *user* or *admin* or some other group or no group. SDT Connector will not auto-configure the *root* (and we recommend that you only use this account for initial config and to add an initial *admin* account to the *console server*).

6.2.4 Make an SDT connection through the gateway to a host

- Simply ***point*** at the host to be accessed ***and click*** on the service to use to access that host. The SSH tunnel to the gateway is then automatically established, the appropriate ports redirected through to the host, and the appropriate local client application is launched pointing at the local endpoint of the redirection:



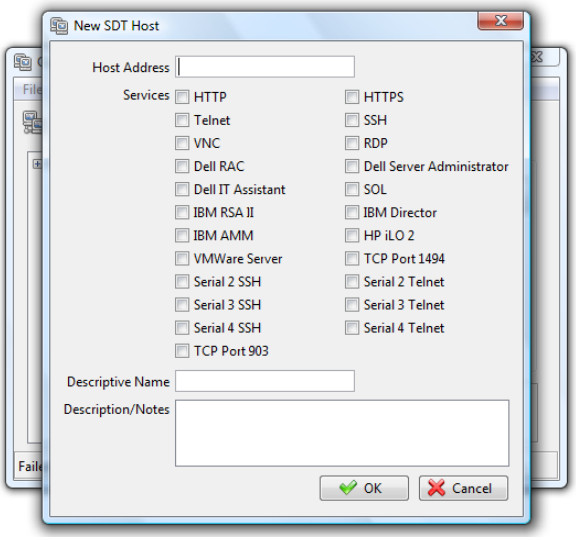
Note You can configure the SDT Connector client can be configured with unlimited number of Gateways (that is, *console servers*). You can configure each Gateway to port forward to an unlimited number of locally networked Hosts. There is no limit on the number of SDT Connector clients that can be configured to access the one Gateway. Nor are there limits on the number of Host connections that an SDT Connector client can concurrently have open through the one Gateway tunnel.

There is a limit on the number of SDT Connector SSH tunnels that can be open at the same time on a particular Gateway *(console server)*. Each Gateway *(console server)* can support at least 50 such concurrent connections. At any time, you could have up to 50 users securely controlling an unlimited number of Managed Devices at a remote site through the on-site *console server* Gateway.

6.2.5 Manually adding hosts to the SDT Connector gateway

For each gateway, you can manually specify the network connected hosts that you will access through that *console server*; and for each host, specify the services that you will use to communicate with the host.

- Select the newly added gateway and click the *Host* icon to create a host that will be accessible via this gateway. (Alternatively select **File: New Host**).

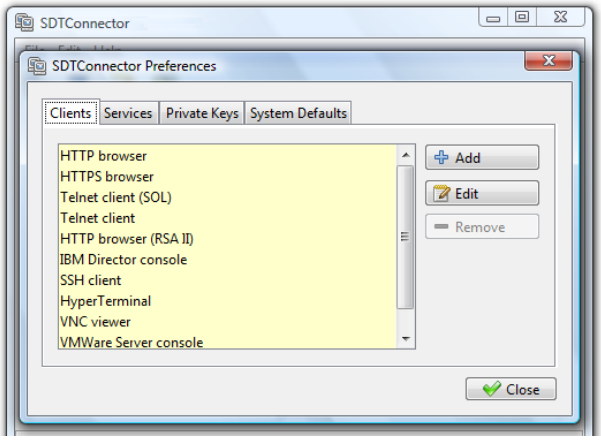


- Enter the IP or DNS **Host Address** of the host (if this is a DNS address, it must be able to be resolved by the gateway).
- Select which **Services** to use to access the new host. A range of service options are pre-configured in the default *SDT Connector* client (RDP, VNC, HTTP, HTTPS, Dell RAC, VMware, etc.). However if you want to add new services to the range, then proceed to the next section (**Adding a new service**) then return here.
- Or, enter a **Descriptive Name** for the host to display instead of the IP or DNS address, and any **Notes** or a **Description** of this host (such as its operating system/release, or anything special about its configuration).
- Click **OK.**

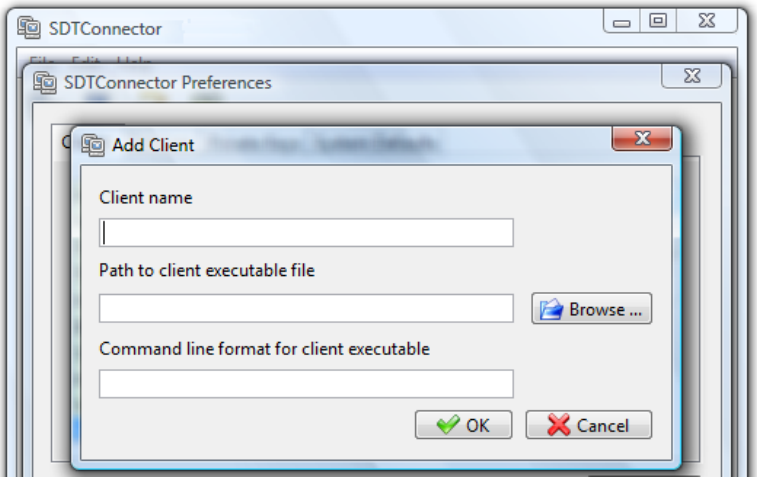
6.2.6 Manually adding new services to the new hosts

To extend the range of services that you can use when accessing hosts with *SDT Connector*:

- Select **Edit: Preferences** and click the **Services** tab. Click **Add.**
- Enter a **Service Name** and click **Add.**
- Under the **General** tab, enter the TCP Port that this service runs on (for example, 80 for HTTP). Or, select the client to use to access the local endpoint of the redirection.



- Select which **Client** application is associated with the new service. A range of client application options are pre-configured in the default *SDT Connector* (RDP client, VNC client, HTTP browser, HTTPS browser, Telnet client, etc.). If you want to add new client applications to this range, proceed to the next section (**Adding a new client**), then return here.

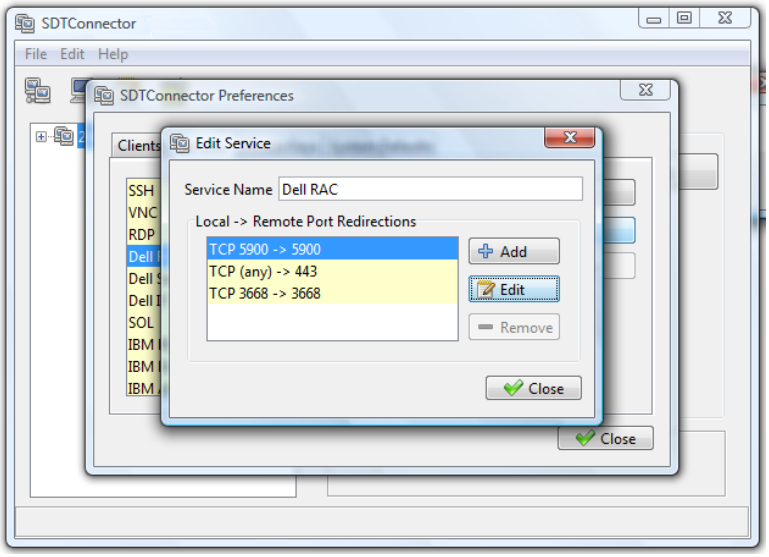


- Click **OK**, then **Close.**

A service typically consists of a single SSH port redirection and a local client to access it. It may consist of several redirections, and some or all may have clients associated with them.

An example is the Dell RAC service. The first redirection is for the HTTPS connection to the RAC server—it has a client associated with it (web browser) that it launches immediately when you click the button for this service.

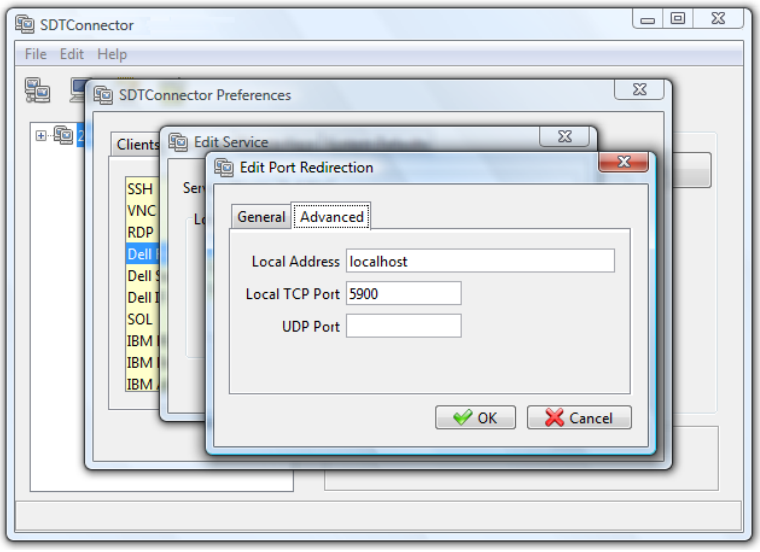
The second redirection is for the VNC service that you may choose to later launch from the RAC web console. It automatically loads in a Java client served through the web browser, so it does not need to have a local client associated with it.



- On the Add Service screen, you can click **Add** as many times as needed to add multiple new port redirections and associated clients.

You may also specify **Advanced** port redirection options:

- Enter the local address to bind to when creating the local endpoint of the redirection. It is not usually necessary to change this from "localhost."
- Enter a local TCP port to bind to when creating the local endpoint of the redirection. If you leave this blank, a random port is selected.



Note *SDT Connector* can also tunnel UDP services. *SDT Connector* tunnels the UDP traffic through the TCP SSH redirection, so it is a "tunnel within a tunnel."

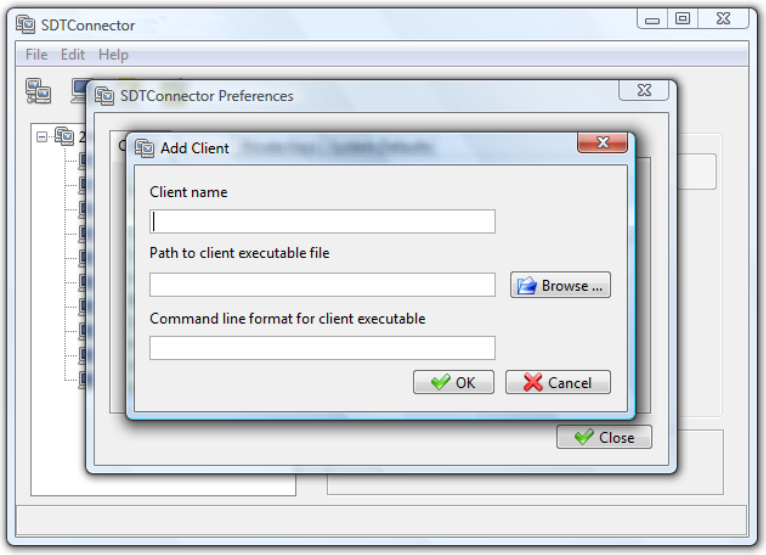
Enter the UDP port where the service is running on the host. This will also be the local UDP port that *SDT Connector* binds as the local endpoint of the tunnel.

Note that for UDP services, you still need to specify a TCP port under General. This will be an arbitrary TCP port that is not in use on the gateway. An example of this is the SOL Proxy service. It redirects local UDP port 623 to remote UDP port 623 over the arbitrary TCP port 6667.

6.2.7 Adding a client program to be started for the new service

Clients are local applications that you may launch when a related service is clicked. To add to the pool of client programs:

- Select **Edit: Preferences** and click the **Client** tab. Click **Add.**



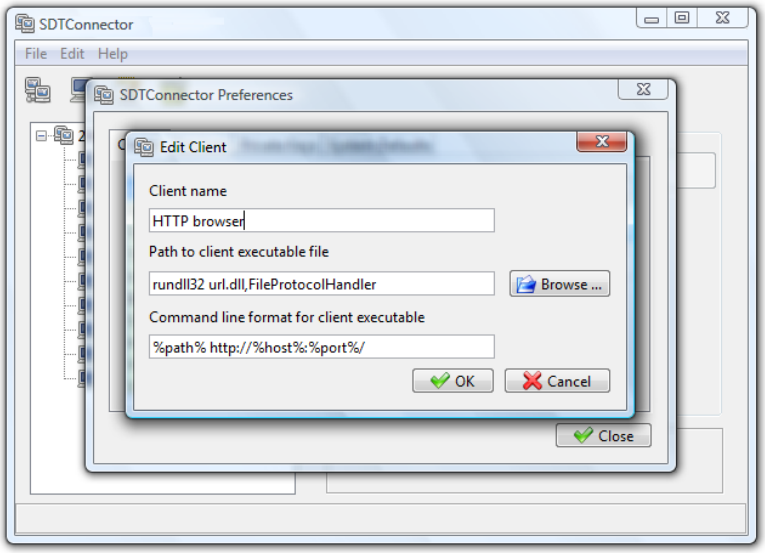
- Enter a **Name** for the client. Enter the **Path** to the executable file for the client (or click **Browse** to locate the executable).
- Enter a **Command Line** associated with launching the client application. *SDT Connector* typically launches a client using command line arguments to point it at the local endpoint of the redirection. There are three special keywords for specifying the command line format. When launching the client, *SDT Connector* substitutes these keywords with the appropriate values:

  %path% is path to the executable file, that is, the previous field.

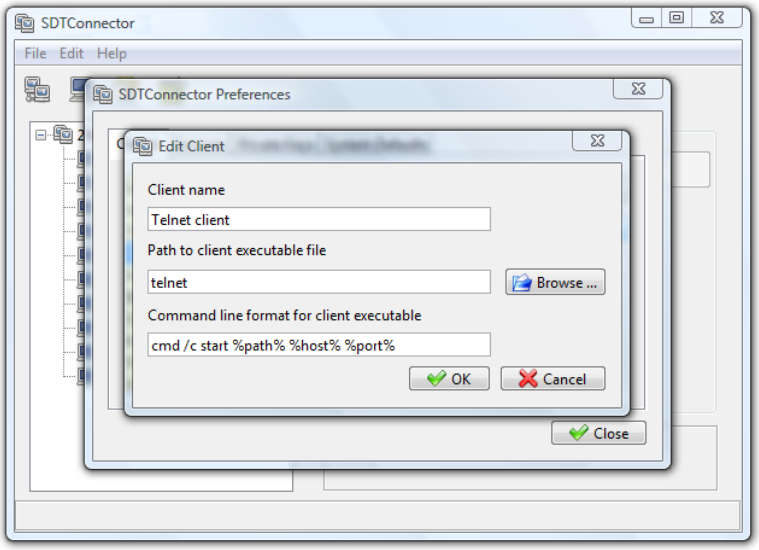
  %host% is the local address to which the local endpoint of the redirection is bound, that is, the Local Address field for the Service redirection Advanced options.

  %port% is the local port to which the local endpoint of the redirection is bound, that is, the Local TCP Port field for the Service redirection Advanced options. If this port is unspecified (that is, "Any"), the appropriate randomly selected port will be substituted.

For example *SDT Connector* is preconfigured for Windows installations with a HTTP service client that will connect with the local browser that the local Windows user has configured as the default. Otherwise, the default browser used is Firefox:



Also some clients are launched in a command line or terminal window. The Telnet client is an example of this so the "Path to client executable file" is *telnet* and the "Command line format for client executable" is *cmd /c start %path% %host% %port%* :



- Click **OK.**

6.2.8 Dial in configuration

If the client PC is dialing into *Local/Console* port on the *console server,* you will need to set up a dial-in PPP link:

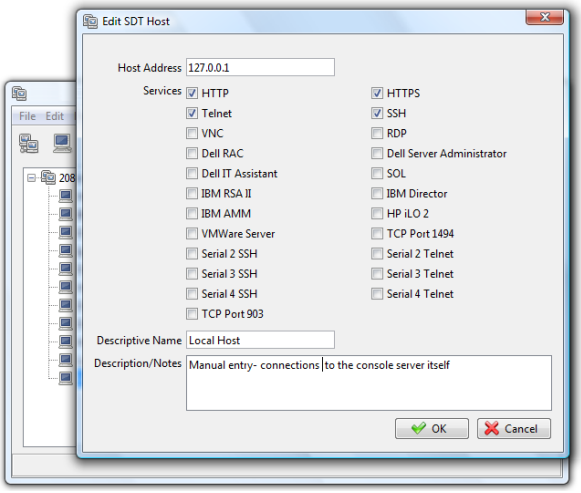
- Configure the *console server* for dial-in access (following the steps in the ***Configuring for Dial-In PPP Access*** section in *Chapter 5*, *Configuring Dial In Access*).
- Set up the PPP client software at the remote *User* PC (following the ***Set up the remote Client*** section in *Chapter 5*).

Once you have a dial-in PPP connection established, you then can set up the secure SSH tunnel from the remote Client PC to the *console server*.

6.3 SDT Connector to Management Console

You can also configure *SDT Connector* for browser access to the *console server's* Management Console —and for Telnet or SSH access to the command line. For these connections to the *console server* itself, you must configure *SDT Connector* to access the Gateway itself by setting the Gateway (*console server)* up as a *host*, and then configuring the appropriate services:

- Launch *SDT Connector* on your PC. Assuming you have already set up the *console server* as a *Gateway* in your *SDT Connector* client (with *username/ password* etc.), select this newly added *Gateway* and click the Host icon to create a host. Or, select **File -> New Host.**
- Enter 127.0.0.1 as the **Host Address** and provide details in **Descriptive Name/Notes**. Click OK.



- Click the **HTTP** or **HTTPS** Services icon to access the Management Console, and/or click **SSH** or **Telnet** to access the command line console.

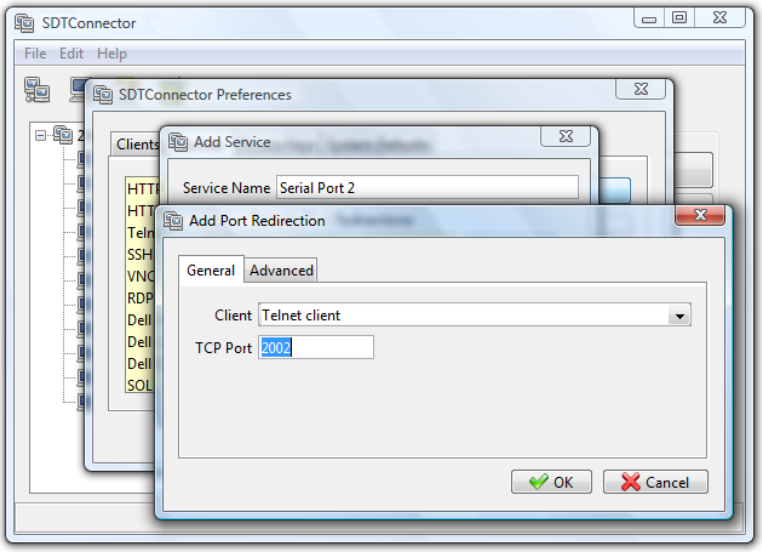
Note: To enable SDT access to the console, you must also configure the *console server* to allow the port forwarded network access to itself:

- Browse to the *console server* and select **Network Hosts** from **Serial & Network**, click **Add Host,** and in the **IP Address/DNS Name** field enter 127.0.0.1 (this is the Black Box network loopback address). Then, enter *Loopback* in **Description**.
- Remove all entries under **Permitted Services** except for those that you will use to access the Management Console (80/http or 443/https) or the command line (22/ssh or 23/telnet). Scroll to the bottom and click **Apply.**
- *Administrators* by default have gateway access privileges. For *Users* to access the *console server* Management Console, you will need to give those *Users* the required access privileges. Select **Users & Groups** from **Serial & Network**. Click **Add User**. Enter a **Username**, **Description** and **Password/Confirm**. Select 127.0.0.1 from **Accessible Host**(s) and click **Apply.**

6.4 SDT Connector - telnet or SSH connect to serially attached devices

You can also use *SDT Connector* to access text consoles on devices that are attached to the *console server* serial ports. For these connections, you must configure the *SDT Connector* client software with a Service that will access the target gateway serial port, and then set the gateway up as a host:

- Launch *SDT Connector* on your PC. Select **Edit -> Preferences** and click the **Services** tab. Click **Add**.
- Enter *"Serial Port 2"* in **Service Name** and click **Add**.
- Select **Telnet** client as the Client. Enter 2002 in **TCP Port**. Click **OK**, then **Close** and **Close** again.



- Assuming you have already set up the target *console server* as a *gateway* in your *SDT Connector* client (with *username/ password* etc), select this *gateway* and click the **Host** icon to create a host. Or, select **File -> New Host**.
- Enter 127.0.0.1 as the **Host Address** and select **Serial Port 2** for Service. In **Descriptive Name**, enter something such as Loopback ports, or Local serial ports. Click **OK.**
- Click *Serial Port 2* icon for Telnet access to the serial console on the device attached to serial port #2 on the gateway.

To enable *SDT Connector* to access to devices connected to the gateway's serial ports, you must also configure the *Console server* itself to allow port forwarded network access to itself, and enable access to the nominated serial port:

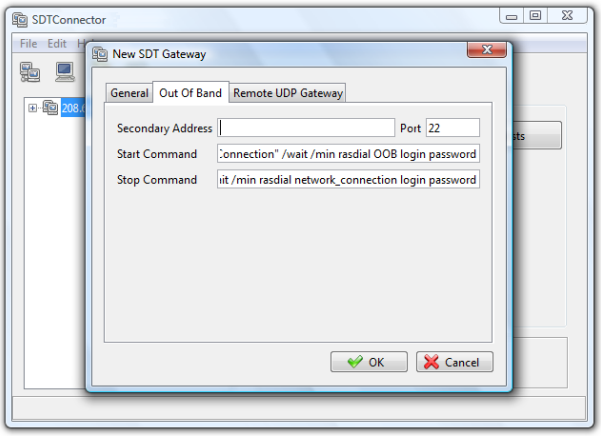
- Browse to the *Console server* and select **Serial Port** from **Serial & Network.**
- Click **Edit** next to selected Port # (for example, Port 2 if the target device is attached to the second serial port). Make sure the port's serial configuration is appropriate for the attached device.
- Scroll down to ***Console server* Setting** and select ***Console server* Mode**. Check **Telnet** (or SSH) and scroll to the bottom and click **Apply.**
- Select **Network Hosts** from **Serial & Network** and click **Add Host.**
- In the **IP Address/DNS Name** field enter *127.0.0.1* (this is the Black Box network loopback address) and enter *Loopback* in **Description**.
- Remove all entries under **Permitted Services**, select **TCP,** and enter *200n* in **Port**. (This configures the Telnet port enabled in the previous step, so for Port 2 you would enter *2002*.)
- Click **Add,** then scroll to the bottom and click **Apply.**
- *Administrator*s by default have gateway and serial port access privileges; however for *Users* to access the gateway and the serial port, you will need to give those *Users* the required access privileges. Select **Users & Groups** from **Serial & Network**. Click **Add User**. Enter a **Username**, **Description,** and **Password/Confirm**. Select 127.0.0.1 from **Accessible Host**(s) and select Port 2 from Accessible Port(s). Click **Apply.**

6.5 Using SDT Connector for out-of-band connection to the gateway

You can also set up *SDT Connector* to connect to the *console server* (gateway) out-of-band (OoB). OoB access uses an alternate path for connecting to the gateway to that used for regular data traffic. OoB access is useful for when the primary link into the gateway is unavailable or unreliable.

Typically, a gateway's primary link is a broadband Internet connection or Internet connection via a LAN or VPN, and the secondary out-of-band connectivity is provided by a dial-up or wireless modem directly attached to the gateway. Out-of-band access enables you to access the hosts and serial devices on the network, diagnose any connectivity issues, and restore the gateway's primary link.

In *SDT Connector*, to configure OoB access, you provide the secondary IP address of the gateway, and tell *SDT Connector* how to start and stop the OoB connection. You can start an OoB connection by initiating a dial up connection, or adding an alternate route to the gateway. *SDT Connector* allows for maximum flexibility. It allows you to provide your own scripts or commands for starting and stopping the OoB connection.



To configure *SDT Connector* for OoB access:

- When adding a new Gateway or editing an existing Gateway select the **Out Of Band** tab.
- Enter the secondary, OoB IP address of the gateway (for example, the IP address it is using when dialed in directly). You also may modify the gateway's SSH port if it's not using the default of 22.
- Enter the command or path to a script to start the OoB connection in **Start Command.**
  - To initiate a pre-configured dial-up connection under Windows, use the following Start Command:

    cmd /c start "Starting Out of Band Connection" /wait /min rasdial network_connection login password

    where *network_connection* is the name of the network connection as displayed in *Control Panel -> Network Connections*, *login* is the dial-in username, and *password* is the dial-in password for the connection.
  - To initiate a pre-configured dial-up connection under Linux, use the following Start Command:

pon network_connection

where *network_connection* is the name of the connection.

- Enter the command or path to a script to stop the OoB connection in **Stop Command**.
  - To stop a pre-configured dial-up connection under Windows, use the following Stop Command:

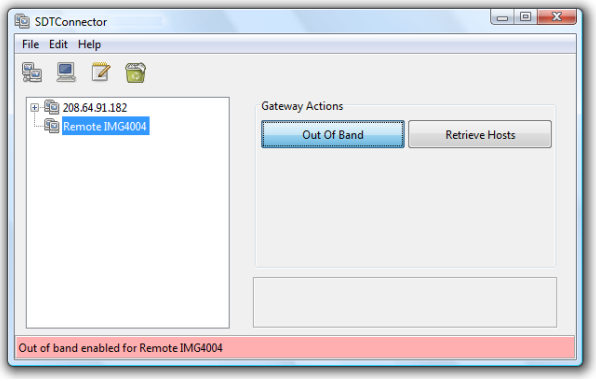
    cmd /c start "Stopping Out of Band Connection" /wait /min rasdial network_connection /disconnect

    where *network connection* is the name of the network connection as displayed in *Control Panel -> Network Connections*.
  - To stop a pre-configured dial-up connection under Linux, use the following Stop Command:

    poff network_connection

To make the OoB connection using *SDT Connector:*

- Select the *console server* and click Out Of Band. The status bar will change color to indicate that this *console server* is now accessed using the OoB link rather than the primary link.



When you connect to a service on a host behind the *console server*, or to the *console server* itself, *SDT Connector* will initiate the OoB connection using the provided Start Command. The OoB connection does not stop (using the provided Stop Command) until you click off Out Of Band under Gateway Actions; then the status bar will return to its normal color.

6.6 Importing (and exporting) preferences

To enable the distribution of pre-configured client config files, *SDT Connector* has an *Export/Import* facility:

- To save a configuration.xml file (for backup or for importing into other *SDT Connector* clients) select **File -> Export Preferences** and select the location where you want to save the configuration file.

- To import a configuration, select **File -> Import Preferences** and select the .xml configuration file to install.

6.7 SDT Connector Public Key Authentication

SDT Connector can authenticate against an SSH gateway using your SSH key pair instead of requiring you to enter your password. This is known as public key authentication.

To use public key authentication with SDT Connector, first you must add the public part of your SSH key pair to your SSH gateway:

- Make sure the SSH gateway allows public key authentication, this is typically the default behavior.
- If you do not already have a public/private key pair for your client PC (the one running SDT Connector), generate them now using *ssh-keygen, PuTTYgen* or a similar tool. You may use RSA or DSA; however, leave the passphrase field blank:
  - PuTTYgen: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
  - OpenSSH: http://www.openssh.org/
  - OpenSSH (Windows): http://sshwindows.sourceforge.net/download/
- Upload the public part of your SSH key pair (this file is typically named *id_rsa.pub* or *id_dsa.pub*) to the SSH gateway, or otherwise add to *.ssh/authorized keys* in your home directory on the SSH gateway.
- Next, add the private part of your SSH key pair (this file is typically named *id_rsa* or *id_dsa*) to SDT Connector. Click **Edit -> Preferences -> Private Keys -> Add**, locate the private key file, and click **OK.**

You do not have to add the public part of your SSH key pair, the private key calculates it.

SDT Connector will now use public key authentication when connecting through the SSH gateway (*console server*). You may have to restart SDT Connector to shut down any existing tunnels that were established using password authentication.

If you have a host behind the *console server* that you connect to by clicking the SSH button in SDT Connector, you may also want to configure access to it for public key authentication as well. This configuration is entirely independent of SDT Connector and the SSH gateway. You must configure the SSH client that SDT Connector launches (for example, Putty, OpenSSH) and the host's SSH server for public key authentication. Essentially what you are using is SSH over SSH, and the two SSH connections are entirely separate.

6.8 Setting up SDT for Remote Desktop access

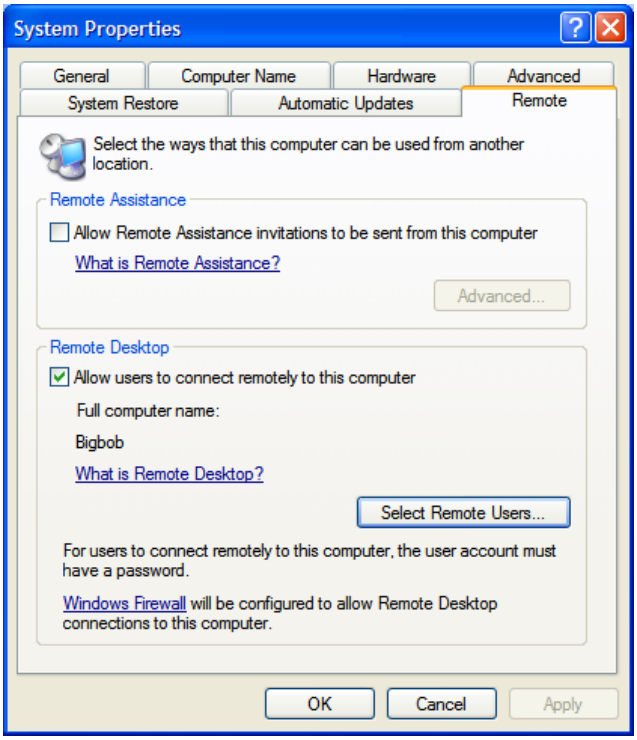
The Microsoft Remote Desktop Protocol (RDP) enables the system manager to securely access and manage remote Windows computers—to reconfigure applications and user profiles, upgrade the server's operating system, reboot the machine, etc. Black Box's Secure Tunneling uses SSH tunneling, so this RDP traffic is securely transferred through an authenticated and encrypted tunnel.

SDT with RDP also allows remote *Users* to connect to Windows XP, Vista, Server2003, and Server 2008 computers and to Windows 2000 Terminal Servers; and to access to all of the applications, files, and network resources (with full graphical interface just as though they were in front of the computer screen at work). To set up a secure Remote Desktop connection, enable Remote Desktop on the target Windows computer that you want to access and configure the RPD client software on the client PC.

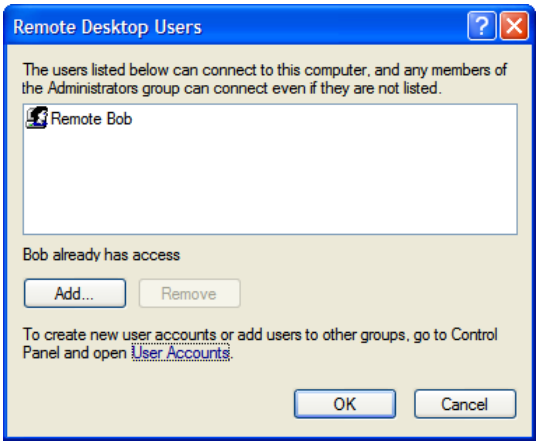
6.8.1 Enable Remote Desktop on the target Windows computer to be accessed

To enable **Remote Desktop** on the Windows computer being accessed:

- Open **System** in the Control Panel and click the **Remote** tab.



- Check **Allow users to connect remotely to this computer**.
- Click **Select Remote Users.**


Remote Desktop Users
The users listed below can connect to this computer, and any members of the Administrators group can connect even if they are not listed.
Remote Bob
Bob already has access
Add...
Remove
To create new user accounts or add users to other groups, go to Control Panel and open User Accounts.
OK
Cancel

- To set the user(s) who can remotely access the system with RDP, click **Add** on the **Remote Desktop Users** dialog box.

Note If you need to set up new users for Remote Desktop access, open **User Accounts** in the Control Panel and follow the steps to nominate the new user's name, password, and account type (*Administrator* or Limited).

Note With Windows XP Professional and Vista, you have only one Remote Desktop session and it connects directly to the Windows root console. With Windows Server 2008, you can have multiple sessions (and with Server 2003 you have three sessions— the console session and two other general sessions). More than one user can have active sessions on a single computer.

When the remote user connects to the accessed computer on the console session, Remote Desktop automatically locks that computer (no other user can access the applications and files). When you come back to your computer at work, you can unlock it by typing CTRL+ALT+DEL.

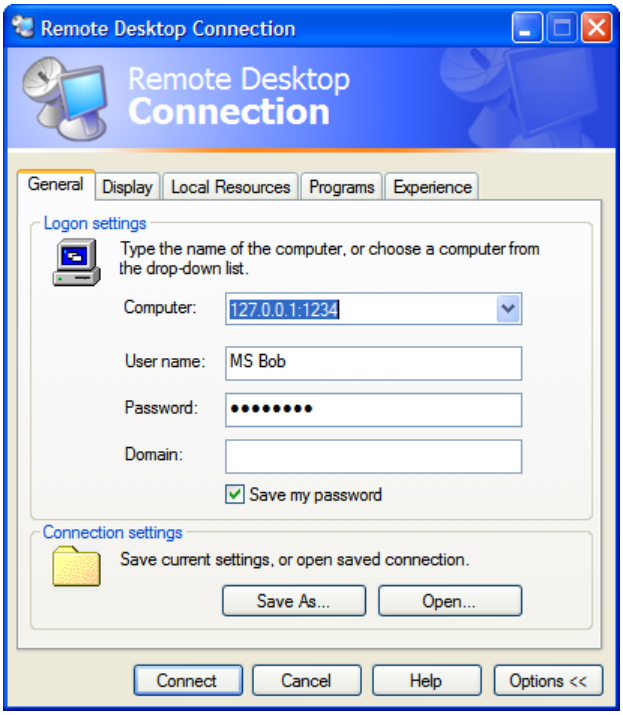
6.8.2 Configure the Remote Desktop Connection client

Now that you have the Client PC securely connected to the *console server* (either locally, or remotely—through the enterprise VPN, or a secure SSH internet tunnel, or a dial-in SSH tunnel), you can establish the Remote Desktop connection from the Client. Simply enable the **Remote Desktop Connection** on the remote client PC, then point it to the SDT Secure Tunnel port in the *console server*:

A. On a Windows client PC

- Click **Start**. Point to **Programs**, then to **Accessories**, then **Communications**, and click **Remote Desktop Connection**.

- In **Computer**, enter the appropriate IP Address and Port Number:
  - Where there is a direct local or enterprise VPN connection, enter the IP Address of the *console server*, and the Port Number of the SDT Secure Tunnel for the *console server* serial port that you attach to the Windows computer you want to control. For example, if the Windows computer is connected to serial Port 3 on a *console server* located at 192.168.0.50, then you would enter *192.168.0.50:7303.*
  - Where there is an SSH tunnel (over a dial up PPP connection or over a public internet connection or private network connection), simply enter the *localhost* as the IP address, *127.0.0.1.* For Port Number, enter the *source port* you created when setting SSH tunneling /port forwarding (in Section 6.1.6), for example, *:1234.*
- Click **Option.** In the **Display** section, specify an appropriate color depth (for example, for a modem connection we recommend that you not use over 256 colors). In **Local Resources,** specify the peripherals on the remote Windows computer that are to be controlled (printer, serial port, etc.).



- Click **Connect.**

Note The Remote Desktop Connection software is pre-installed with Windows XP, Vista and Server 2003/2008. For earlier Windows PCs, you need to download the RDP client:

- Go to the Microsoft Download Center site http://www.microsoft.com/downloads/details.aspx?familyid=80111F21-D48D-426E-96C2-08AA2BD23A49&displaylang=en and click the **Download** button

  This software package will install the client portion of Remote Desktop on Windows 95, Windows 98 and 98 Second Edition, Windows Me, Windows NT 4.0, and Windows 2000. When run, this software allows these older Windows platforms to remotely connect to a computer running current Windows.

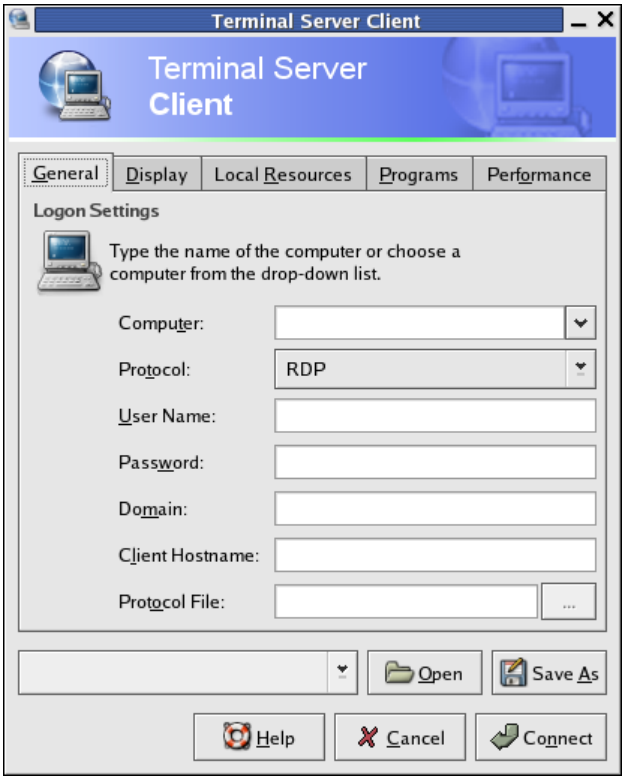
B. On a Linux or UNIX client PC:

- Launch the open source *rdesktop* client:

  rdesktop -u *windows-user-id* -p *windows-password* -g 1200x950 *ms-windows-terminal-server-host-name*

option	description
-a	Color depth: 8, 16, 24
-r	Device redirection. (Redirect sound on remote machine to local device. -0 -r sound (MS/Windows 2003)
-g	Geometry: *widthxheight* or 70% screen percentage.
-p	Use -p - to receive password prompt.

- You can use GUI front end tools like the GNOME Terminal Services Client *tsclient* to configure and launch the *rdesktop* client. (Using *tsclient* also enables you to store multiple configurations of *rdesktop* for connection to many servers)



Note The *rdesktop* client is supplied with Red Hat 9.0:

- rpm -ivh rdesktop-1.2.0-1.i386.rpm

For Red Hat 8.0 or other distributions of Linux; download source, untar, configure, make, make, then install.

rdesktop currently runs on most UNIX based platforms with the X Window System and can be downloaded from http://www.rdesktop.org/

C. On a Macintosh client:

- Download Microsoft's free Remote Desktop Connection client for Mac OS X http://www.microsoft.com/mac/otherproducts/otherproducts.aspx?pid=remotedesktopclient

6.9 SDT SSH Tunnel for VNC

With SDT and Virtual Network Computing (VNC), *Users* and *Administrators* can securely access and control Windows 98/NT/2000/XP/2003, Linux, Macintosh, Solaris, and UNIX computers. There's a range of popular free and commercial VNC software available (UltraVNC, RealVNC, TightVNC). To set up a secure VNC connection, install and configure the VNC Server software on the computer the user will access, then install and configure the VNC Viewer software on the Viewer PC.

6.9.1 Install and configure the VNC Server on the computer to be accessed

Virtual Network Computing (VNC) software enables users to remotely access computers running Linux, Macintosh, Solaris, UNIX, all versions of Windows, and most other operating systems.

A. For Microsoft Windows servers (and clients):

Windows does not include VNC software, so you will need to download, install, and activate a third party VNC Server software package:

RealVNC http://www.realvnc.com is fully cross-platform, so a desktop running on a Linux machine may be displayed on a Windows PC, on a Solaris machine, or on any number of other architectures. There is a Windows server, allowing you to view the desktop of a remote Windows machine on any of these platforms using exactly the same viewer. RealVNC was founded by members of the AT&T team who originally developed VNC.

TightVNC http://www.tightvnc.com is an enhanced version of VNC. It has added features such as file transfer, performance improvements, and read-only password support. They have just recently included a video drive much like UltraVNC. TightVNC is still free, cross-platform (Windows Unix, and Linux), and compatible with the standard (Real) VNC.

UltraVNC http://ultravnc.com is easy to use, fast, and free VNC software that has pioneered and perfected features that the other flavors have consistently refused or been very slow to implement for cross platform and minimalist reasons. UltraVNC runs under Windows operating systems (95, 98, Me, NT4, 2000, XP, 2003). Download UltraVNC from Sourceforge's UltraVNC file list.

B. For Linux servers (and clients):

Most Linux distributions now include VNC Servers and Viewers and they generally can be launched from the (Gnome/KDE etc) front end; for example, with Red Hat Enterprise Linux 4 there's VNC Server software and a choice of Viewer client software, and to launch:

- Select the **Remote Desktop** entry in the **Main Menu -> Preferences** menu.
- Click the **Allow other users…** checkbox to allow remote users to view and control your desktop.

Remote Desktop Preferences
Sharing
Allow other users to view your desktop
Allow other users to control your desktop
Users can view your desktop using this command:
vncviewer hoopoe.elk:0
Security
When a user tries to view or control your desktop:
Ask you for confirmation
Require the user to enter this password:
Password:
Close

- To set up a persistent VNC server on Red Hat Enterprise Linux 4:
  - Set a password using **vncpasswd**
  - Edit **/etc/sysconfig/vncservers**
  - Enable the service with **chkconfig vncserver on**
  - Start the service with **service vncserver start**
  - Edit **/home/*username*/.vnc/xstartup** if you want a more advanced session than just *twm* and an xterm.

C. For Macintosh servers (and clients):

OSXvnc http://www.redstonesoftware.com/vnc.html is a robust, full-featured VNC server for Mac OS X that allows any VNC client to remotely view and/or control the Mac OS X machine. OSXvnc is supported by Redstone Software.

D. Most other operating systems (Solaris, HPUX, PalmOS etc) either come with VNC bundled, or have third-party VNC software that you can download.

6.9.2 Install, configure and connect the VNC Viewer

VNC is truly *platform-independent* so a VNC Viewer on any operating system can connect to a VNC Server on any other operating system. There are Viewers (and Servers) from a wide selection of sources (for example, UltraVNC TightVNC or RealVNC) for most operating systems. There are also a wealth of Java viewers available so that any desktop can be viewed with any Java-capable browser (http://en.wikipedia.org/wiki/VNC lists many of the VNC Viewers sources).

- Install the VNC Viewer software and set it up for the appropriate speed connection.

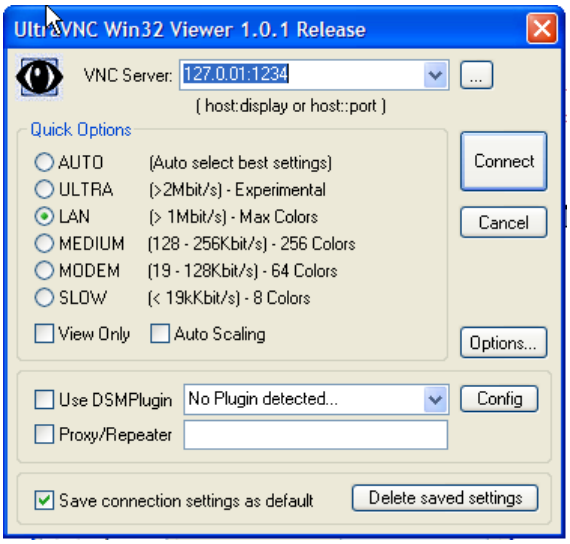
Note To make VNC faster, when you set up the Viewer:

- Set encoding to ZRLE (if you have a fast enough CPU).
- Decrease color level (e.g. 64 bit).
- Disable the background transmission on the Server or use a plain wallpaper.

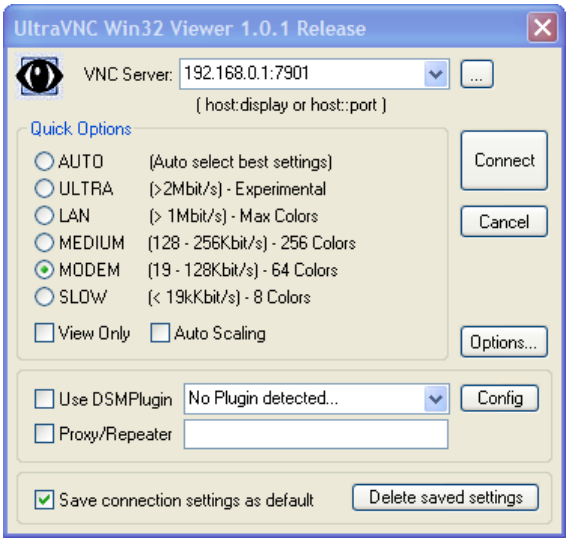
(Refer to http://doc.uvnc.com for detailed configuration instructions)

- To establish the VNC connection, first configure the VNC Viewer, entering the VNC Server IP address.

A. When the Viewer PC is connected to the *console server* thru an SSH tunnel (over the public Internet, or a dial-in connection, or private network connection), enter *localhost* (or 127.0.0.1) as the IP VNC Server IP address; and *the source port* you entered when setting SSH tunneling /port forwarding (in Section 6.2.6) e.g. *:1234*



B. When the Viewer PC is connected directly to the *console server* (i.e. locally or remotely through a VPN or dial in connection); and the VNC Host computer is serially connected to the *console server;* enter the IP address of the *console server* unit with the TCP port that the SDT tunnel will use. The TCP port will be 7900 plus the physical serial port number (*i.e.* 7901 to 7948, so all traffic directed to port 79xx on the *console server* is tunneled thru to port 5900 on the PPP connection on serial Port xx). For a Windows Viewer PC using UltraVNC connecting to a VNC Server attached to Port 1 on a *console server*, it is located at 192.168.0.1



- To establish the VNC connection, simply activate the VNC Viewer software on the Viewer PC and enter the password.

Note For general background reading on Remote Desktop and VNC access we recommend the following:

- *The Microsoft Remote Desktop How-To.*
- http://www.microsoft.com/windowsxp/using/mobility/getstarted/remoteintro.mspx
- *The Illustrated Network Remote Desktop* help page. http://theillustratednetwork.mvps.org/RemoteDesktop/RemoteDesktopSetupandTroubleshooting.html
- *What is Remote Desktop in Windows XP and Windows Server 2003?* by Daniel Petri. http://www.petri.co.il/what's_remote_desktop.htm
- *Frequently Asked Questions about Remote Desktop.* http://www.microsoft.com/windowsxp/using/mobility/rdfaq.mspx
- *Secure remote access of a home network using SSH, Remote Desktop and VNC for the home user* http://theillustratednetwork.mvps.org/RemoteDesktop/SSH-RDP-VNC/RemoteDesktopVNCandSSH.html
- *Taking your desktop virtual with VNC,* Red Hat magazine. http://www.redhat.com/magazine/006apr05/features/vnc/ and http://www.redhat.com/magazine/007may05/features/vnc/
- *Wikipedia* general background on VNC http://en.wikipedia.org/wiki/VNC.

6.10 Using SDT to IP connect to hosts that are serially attached to the gateway

Network (IP) protocols like RDP, VNC and HTTP can also be used for connecting to host devices that are serially connected through their COM port to the *console server*. To do this you must:

- establish a PPP connection (Section 6.7.1) between the host and the gateway, then
- set up Secure Tunneling—Ports on the *console server* (Section 6.7.2), then
- configure *SDT Connector* to use the appropriate network protocol to access IP consoles on the host devices that are attached to the *Console server* serial ports (Section 6.7.3)

6.10.1 Establish a PPP connection between the host COM port and *console server*

(This step is only necessary for serially connected computers)

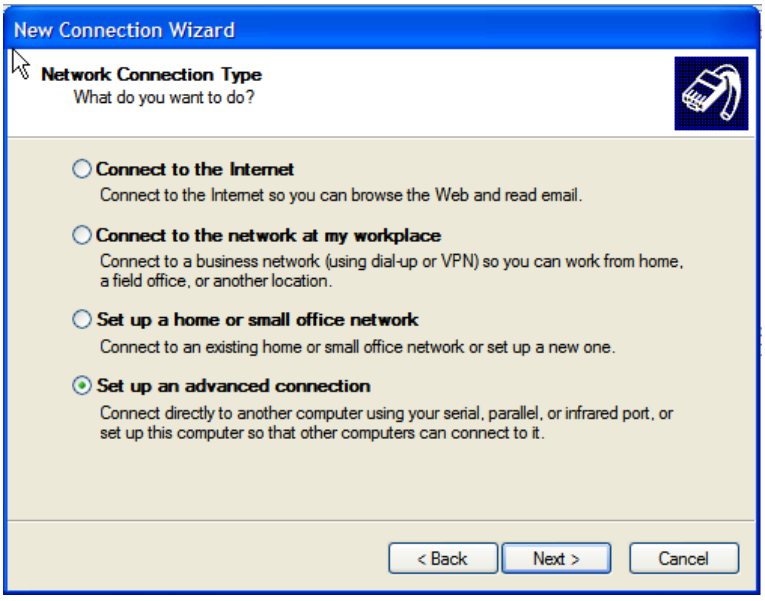
First, physically connect the COM port on the host computer you want to access to the serial port on the *console server,* then:

A. For non Windows (Linux, UNIX, Solaris, etc.) computers, establish a PPP connection over the serial port. The online tutorial http://www.yolinux.com/TUTORIALS/LinuxTutorialPPP.html presents a selection of methods for establishing a PPP connection for Linux.

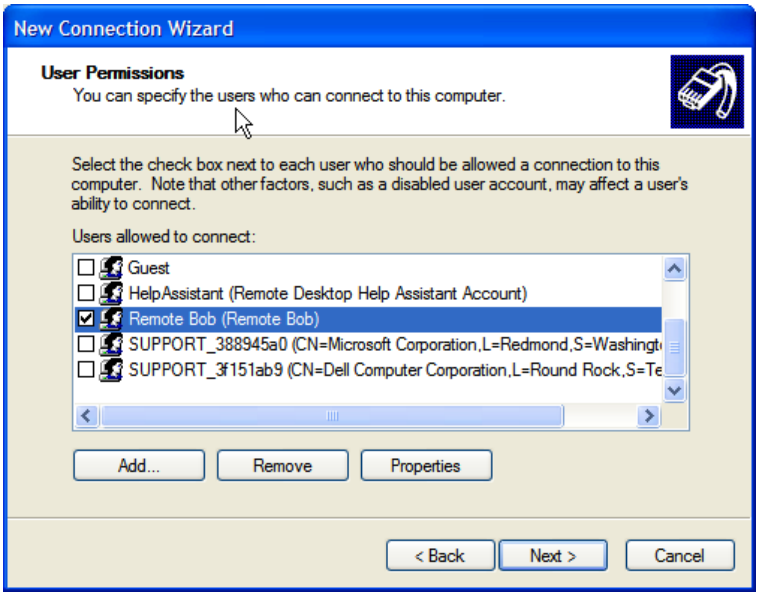
B. For Windows XP and 2003 computers, follow the steps below to set up an advanced network connection between the Windows computer, through its COM port to the *console server*. Both

Windows 2003 and Windows XP Professional allow you to create a *simple dial in service* which can be used for the Remote Desktop/VNC/HTTP/X connection to the *console server*:

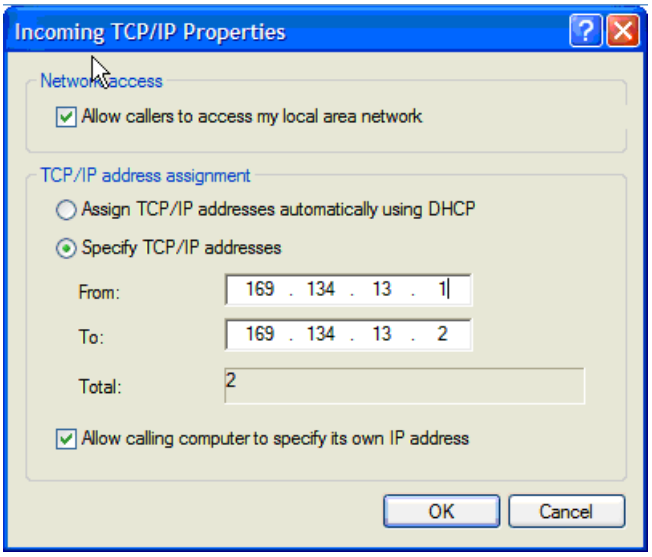
- Open **Network Connections** in Control Panel and click the **New Connection Wizard.**


New Connection Wizard
Network Connection Type
What do you want to do?
Connect to the Internet
Connect to the Internet so you can browse the Web and read email.
Connect to the network at my workplace
Connect to a business network (using dial-up or VPN) so you can work from home, a field office, or another location.
Set up a home or small office network
Connect to an existing home or small office network or set up a new one.
Set up an advanced connection
Connect directly to another computer using your serial, parallel, or infrared port, or set up this computer so that other computers can connect to it.
< Back
Next >
Cancel

- Select **Set up an advanced connection** and click **Next.**
- On the **Advanced Connection Options** screen, select **Accept Incoming Connections** and click **Next.**
- Select the **Connection Device** (i.e. the serial COM port on the Windows computer that you cabled through to the *console server*). By default, select **COM1**. The COM port on the Windows computer should be configured to its maximum baud rate. Click **Next.**
- On the **Incoming VPN Connection Options** screen, select **Do not allow virtual private connections** and click **Next.**


New Connection Wizard
User Permissions
You can specify the users who can connect to this computer.
Select the check box next to each user who should be allowed a connection to this computer. Note that other factors, such as a disabled user account, may affect a user's ability to connect.
Users allowed to connect:
Guest
HelpAssistant (Remote Desktop Help Assistant Account)
Remote Bob (Remote Bob)
SUPPORT_388945a0 (CN=Microsoft Corporation,L=Redmond,S=Washingt
SUPPORT_3f151ab9 (CN=Dell Computer Corporation,L=Round Rock,S=Te
Add...
Remove
Properties
< Back
Next >
Cancel

- Specify which *Users* will be allowed to use this connection. This should be the same *Users* who were given Remote Desktop access privileges in the earlier step. Click **Next.**
- On the **Network Connection** screen select **TCP/IP** and click **Properties.**



- Select **Specify TCP/IP addresses** on the **Incoming TCP/IP Properties** screen, select **TCP/IP.** Nominate a *From:* and a *To:* TCP/IP address, and click **Next.**

Note You can choose any TCP/IP addresses so long as they are addresses that are not used anywhere else on your network. The *From:* address will be assigned to the Windows XP/2003 computer and the *To:* address will be used by the *console server*. For simplicity, use the IP address as shown in the illustration above:

From: 169.134.13.1

To: 169.134.13.2

Or, you can set the advanced connection and access on the Windows computer to use the *console server* defaults:

- Specify 10.233.111.254 as the *From:* address
- Select *Allow calling computer to specify its own address*

Also, you could use the *console server* default username and password when you set up the new Remote Desktop *User* and gave this *User* permission to use the advance connection to access the Windows computer:

- The *console server* default *Username is portXX* where XX is the serial port number on the *console server*.
- The default *Password is portXX*

To use the defaults for a RDP connection to the serial port 2 on the *console server*, you would have set up a Windows user named *port02.*

- When the PPP connection has been set up, a network icon will appear in the Windows task bar.

Note The above notes describe setting up an incoming connection for Windows XP. The steps are similar for Vista and Windows Server 2003/2008, but the set up screens present slightly differently:

You need to put a check in the box for *Always allow directly connected devices such as palmtop…..*

The option for to **Set up an advanced connection** is not available in Windows 2003 if RRAS is configured. If RRAS has been configured, you can enable the null modem connection for the dial-in configuration.

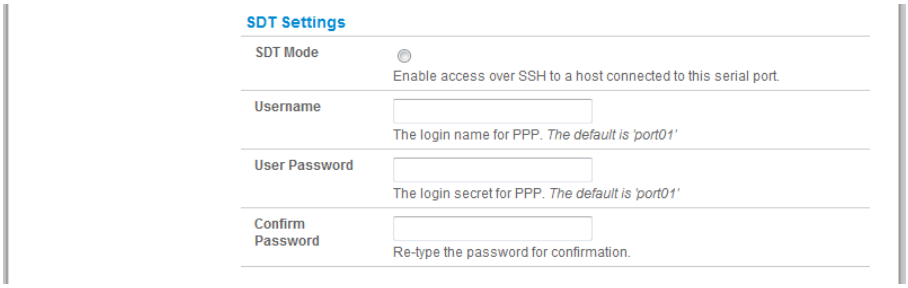
C. For earlier version Windows computers, follow the steps in Section B. above. To get to the **Make New Connection** button:

- For Windows 2000, click **Start,** and select **Settings**. At the **Dial-Up Networking Folder,** click **Network and Dial-up Connections,** and click **Make New Connection.** You may need to first set up a connection over the COM port using **Connect directly to another computer** before proceeding to **Set up an advanced connection.**
- For Windows 98, double click **My Computer** on the Desktop, then open **Dial-Up Networking** and double click.

6.10.2 Set up SDT Serial Ports on *console server*

To set up *RDP (and VNC) forwarding* on the *console server* Serial Port that is connected to the Windows computer COM port:

- Select the **Serial & Network: Serial Port** menu option and click **Edit** (for the particular Serial Port that is connected to the Windows computer COM port).
- On the SDT Settings menu, select **SDT Mode** (this will enable port forwarding and SSH tunneling) and enter a **Username** and **User Password.**


SDT Settings
SDT Mode
Enable access over SSH to a host connected to this serial port.
Username
The login name for PPP. The default is 'port01'
User Password
The login secret for PPP. The default is 'port01'
Confirm Password
Re-type the password for confirmation.

Note When you enable SDT, it will override all other Configuration protocols on that port.

Note If you leave the *Username* and *User Password* fields blank, they default to *portXX* and *portXX* where XX is the serial port number. The default username and password for Secure RDP over Port 2 is *port02.*

- Make sure the *console server* **Common Settings** (Baud Rate, Flow Control) are the same as those set up on the Windows computer COM port and click **Apply.**
- RDP and VNC forwarding over serial ports is enabled on a Port basis. You can add *Users* who can have access to these ports (or reconfigure *User* profiles) by selecting **Serial & Network: User & Groups** menu tag—as described earlier in Chapter 4, *Configuring Serial Ports.*

6.10.3 Set up SDT Connector to SSH port forward over the *console server* Serial Port

In the *SDT Connector* software running on your remote computer, specify the gateway IP address of your *console server* and a username/password for a user you set up on the *console server* that has access to the desired port.

Next, add a New SDT Host. In the Host address, put portxx, where xx = the port you are connecting to. Example: for port 3 you would have a Host Address of: port03. Then select the RDP Service check box.

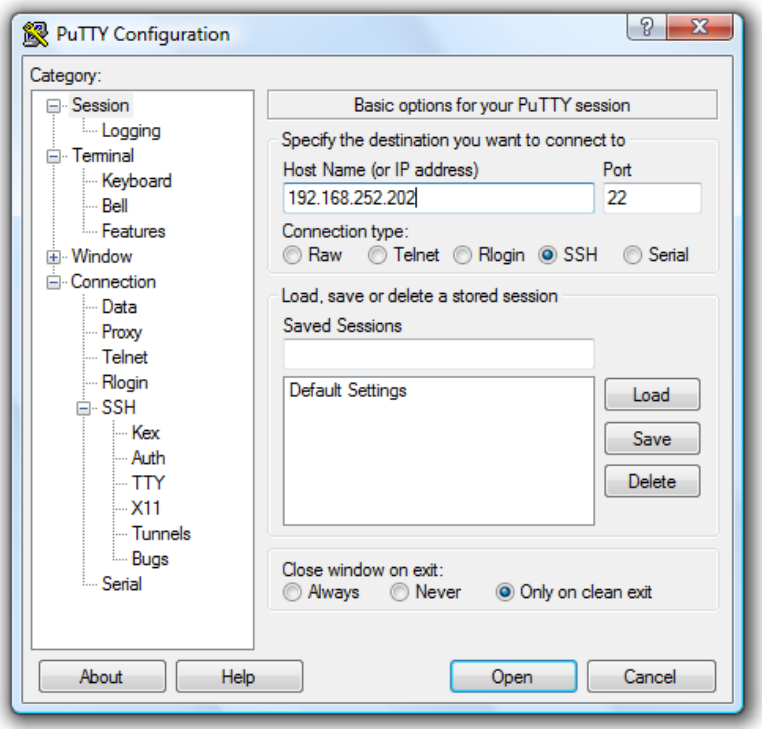
6.11 SSH Tunneling using other SSH clients (e.g. PuTTY)

As covered in the previous sections of this chapter, we recommend that you use the *SDT Connector* client software that is supplied with the *console server*. There's also a wide selection of commercial and free SSH client programs that can provide the secure SSH connections to the *console server*s and secure tunnels to connected devices:

- PuTTY is a complete (though not very user friendly) freeware implementation of SSH for Win32 and UNIX platforms.
- SSHTerm is a useful open source SSH communications package.

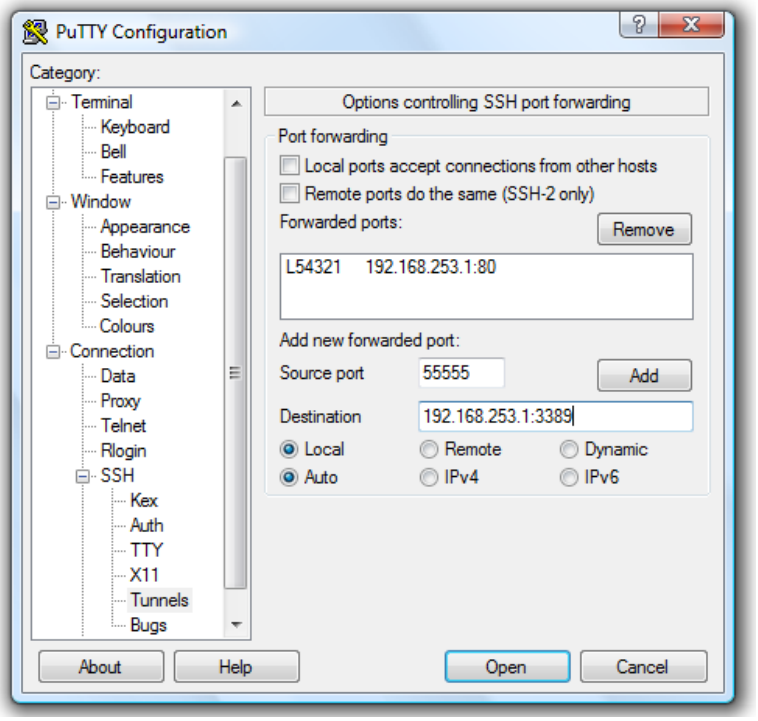
- SSH Tectia is leading end-to-end commercial communications security solution for the enterprise.
- Reflection for Secure IT (formerly F-Secure SSH) is another good commercial SSH-based security solution.

For example, the steps below show how to establish an SSH tunneled connection to a network connected device using the PuTTY client software.



- In the **Session** menu, enter the IP address of the *console server* in the **Host Name or IP address** field.
  - For dial-in connections, this IP address will be the **Local** Address that you assigned to the *console server* when you set it up as the Dial-In PPP Server.
  - For Internet (or local/VPN connections) connections, this will be the *console server's* public IP address.
- Select the **SSH Protocol**, and the **Port** will be set as 22.
- Go to the **SSH -> Tunnels** menu and in *Add new forwarded port* enter any high unused port number for the **Source port**, for example, *54321.*
- Set the **Destination:** IP details.
  - If your destination device is network-connected to the *console server* and you are connecting using RDP, set the Destination as *<Managed Device IP address/DNS Name>:3389*. For example, if when setting up the Managed Device as *Network Host* on the *console server* you specified its IP address to be 192.168.253.1 (or its DNS Name was *accounts.myco.intranet.com*), then specify the Destination as *192.168.523.1:3389* (or *accounts.myco.intranet.com:3389).* Only devices that are configured as networked Hosts can

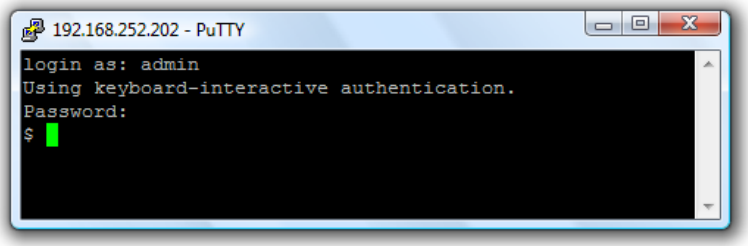
be accessed using SSH tunneling (except by the “root” user who can tunnel to any IP address the *console server* can route to).



- If your destination computer is serially connected to the *console server*, set the *Destination* as *<port label>:3389*. For example, if the **Label** you specified on the serial port on the *console server* is *win2k3*, then specify the remote host as *win2k3:3389*. Or, you can set the *Destination* as *portXX:3389* (where XX is the SDT enabled serial port number). For example, if port 4 is on the *console server* is to carry the RDP traffic, then specify *port04:3389*

Note http://www.jfitz.com/tips/putty_config.html has useful examples on configuring PuTTY for SSH tunneling.

- Select **Local** and click the **Add** button.
- Click **Open** to SSH connect the Client PC to the *console server*. You will now be prompted for the Username/Password for the *console server* user.



- If you are connecting as a *User* in the “users” group, then you can only SSH tunnel to Hosts and Serial Ports where you have specific access permission.

- If you are connecting as an *Administrator* (in the "admin" group), then you can connect to any configured Host or Serial Ports (that has SDT enabled).

To set up the secure SSH tunnel for a HTTP browser connection to the Managed Device, specify port 80 (instead of port 3389 that was used for RDP) in the Destination IP address.

To set up the secure SSH tunnel from the Client (Viewer) PC to the *console server* for VNC, follow the steps above, but when you configure the VNC port redirection, specify port 5900 in the Destination IP address.

Note How secure is VNC? VNC access generally allows access to your whole computer, so security is very important. VNC uses a random challenge-response system to provide the basic authentication that allows you to connect to a VNC server. This is reasonably secure and the password is not sent over the network.

Once connected, all subsequent VNC traffic is unencrypted. A malicious user could snoop your VNC session. There are also VNC scanning programs available, which will scan a subnet looking for PCs that are listening on one of the ports that VNC uses.

Tunneling VNC over a SSH connection ensures all traffic is strongly encrypted. No VNC port is ever open to the internet, so anyone scanning for open VNC ports will not be able to find your computers. When tunneling VNC over a SSH connection, the only port that you're opening on your *console server* is the SDT port 22.

Sometimes it may be prudent to tunnel VNC through SSH even when the Viewer PC and the *console server* are both on the same local network.

Chapter 7 Alerts and Logging

Introduction

This chapter describes the alert generation and logging features of the *console server*. The Alert facility monitors the serial ports, all logins, the power status, and environmental monitors and probes, and sends emails, SMS, Nagios, or SNMP alerts when specified trigger events occur.

- First, enable and configure the service that will be used to carry the alert (*Section 7.1*).
- Then, specify the alert trigger condition and the actual destination to which that particular alert will be sent (*Section 7.2*).

All *console server* models can maintain log records of all access and communications with the *console server* and with the attached serial devices. A log of all system activity is also maintained, as is a history of the status of any attached environmental monitors.

Some models also log access and communications with network attached hosts and maintain a history of the UPS and PDU power status.

- If port logs are to be maintained on a remote server, then configure the access path to this location *(Section 7.3).*
- Then you need to activate and set the desired levels of logging for each serial *(Section 7.4)* and/or network port *(Section 7.5)* and/or power and environment UPS (refer to *Chapter 8*).

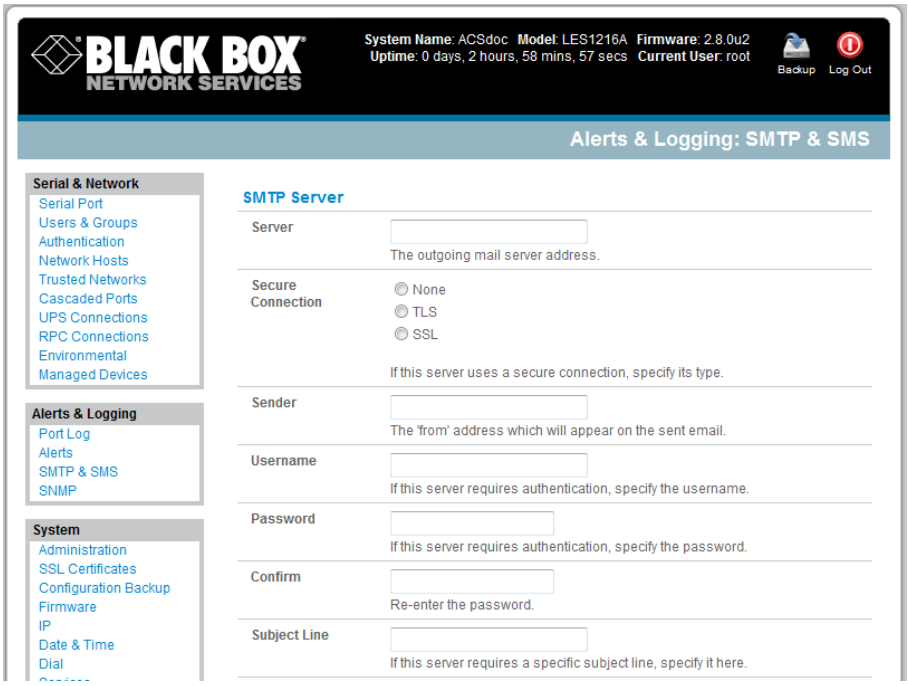
7.1 Configure SMTP/SMS/SNMP/Nagios alert service

The Alerts facility monitors nominated ports/hosts/UPSs/PDUs/EMDs, etc. for trigger conditions. When triggered, the facility sends an alert notification over the nominated alert service. Before setting up the alert trigger, configure these alert services:

7.1.1 Email alerts

The *console server* uses SMTP (Simple Mail Transfer Protocol) for sending the email alert notifications. To use SMTP, the *Administrator* must configure a valid SMTP server for sending the email:

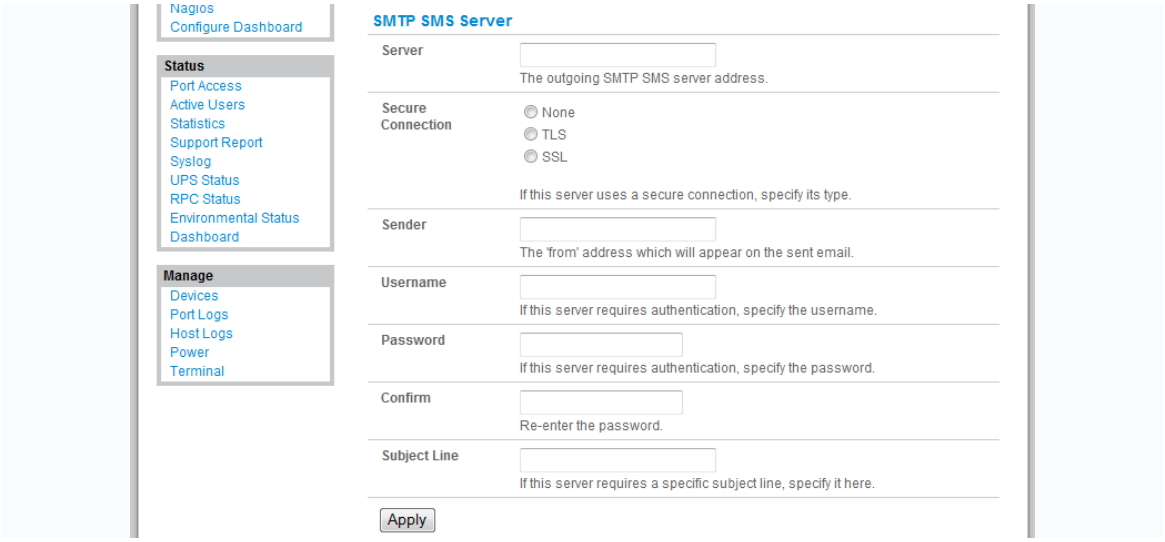
- Select **Alerts & Logging: SMTP &SMS**



- In the **SMTP Server** field, enter the outgoing mail **Server**'s IP address**.**
- If this mail server uses a **Secure Connection**, specify its type.
- You may enter a **Sender** email address which will appear as the "*from"* address in all email notifications sent from this *console server*. Many SMTP servers check the sender's email address with the host domain name to verify the address as authentic. So it may be useful to assign an email address for the console server such as *consoleserver2@mydomain.com*
- You may also enter a **Username** and **Password** if the SMTP server requires authentication.
- You can specify the specific **Subject Line** that will be sent with the email.
- Click **Apply** to activate SMTP.

7.1.2 SMS alerts

The *console server* uses email-to-SMS services to send SMS alert notifications to mobile devices. Sending SMS via email using SMTP (Simple Mail Transfer Protocol) is much faster than sending text pages via a modem using the TAP Protocol. Almost all mobile phone carriers provide an SMS gateway service that forwards email to mobile phones on their networks. There's also a wide selection of SMS gateway aggregators that provide email to SMS forwarding to phones on any carriers. To use SMTP SMS, the *Administrator* must configure a valid SMTP server for sending the email:



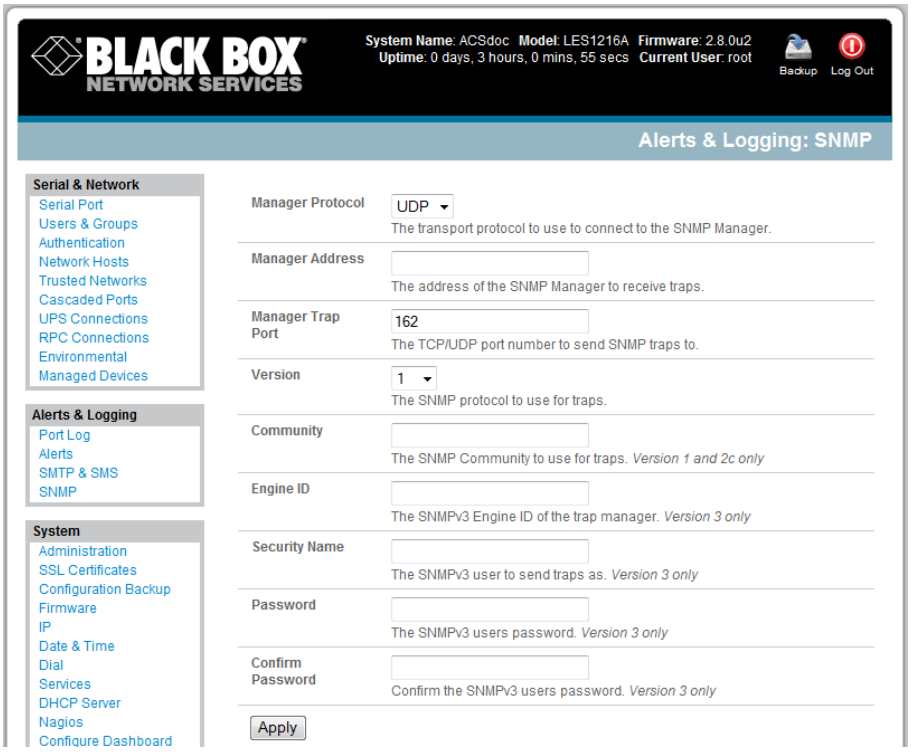
- In the **SMTP SMS Server** field in the **Alerts & Logging: SMTP &SMS** menu, enter the IP address of the outgoing mail **Server** (and **Secure Connection** if applicable).
- You may enter a **Sender** email address, which will appear as the "*from*" address in all email notifications sent from this *console server*. Some SMS gateway service providers only forward email to SMS when the email has been received from authorized senders. You might need to assign a specific authorized email address for the console server.
- You may also enter a **Username** and **Password**, because some SMS gateway service providers use SMTP servers which require authentication.
- You can specify the specific **Subject Line** that will be sent with the email. Generally, the email subject will contain a truncated version of the alert notification message (which is contained in full in the body of the email). However some SMS gateway service providers require blank subjects or require specific authentication headers to be included in the subject line.
- Click **Apply** to activate SMTP.

7.1.3 SNMP alerts

The *Administrator* can configure the Simple Network Management Protocol (SNMP) agent that resides on the *console server* to send SNMP trap alerts to an NMS management application:

- Select **Alerts & Logging: SNMP**
- Enter the SNMP transport protocol. SNMP is generally a **UDP**-based protocol, though infrequently, it uses **TCP** instead.
- Enter the IP address of the **SNMP Manager** and the Port to use for connecting (default = 162)
- Select the version being used. The *console server* SNMP agent supports SNMP v1, v2, and v3.
- Enter the **Community** name for SNMP v1 or 2c. An SNMP community is the group that devices and management stations running SNMP belong to. It helps define where information is sent. SNMP default communities are ***private*** for Write (and public for Read).

- To configure for SNMP v3, you will need to enter an ID and authentication password and contact information for the local *Administrator* (in the **Security Name**).
- Click **Apply** to activate SNMP.



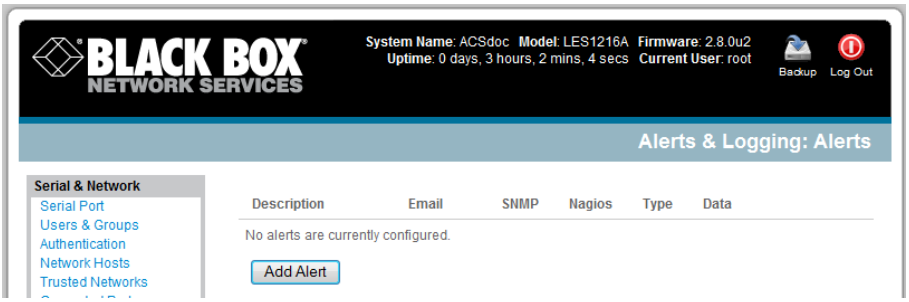
Note All *console servers* have the *snmptrap* daemon to send traps/notifications to remote SNMP servers on defined trigger events as detailed above. LES1208A, LES1216A, and LES1248A *console servers* also embed the *net-snmpd* daemon. It accepts SNMP requests from remote SNMP management servers and provides information on network interface, running processes, etc. (refer to *Chapter 15.5—Modifying SNMP Configuration* for more details).

7.1.4 Nagios alerts

To notify the central Nagios server of Alerts, NSCA must be enabled under **System: Nagios** and Nagios must be enabled for each applicable host or port under **Serial & Network: Network Hosts** or **Serial & Network: Serial Ports** *(refer to Chapter 10).*

7.2 Activate Alert Events and Notifications

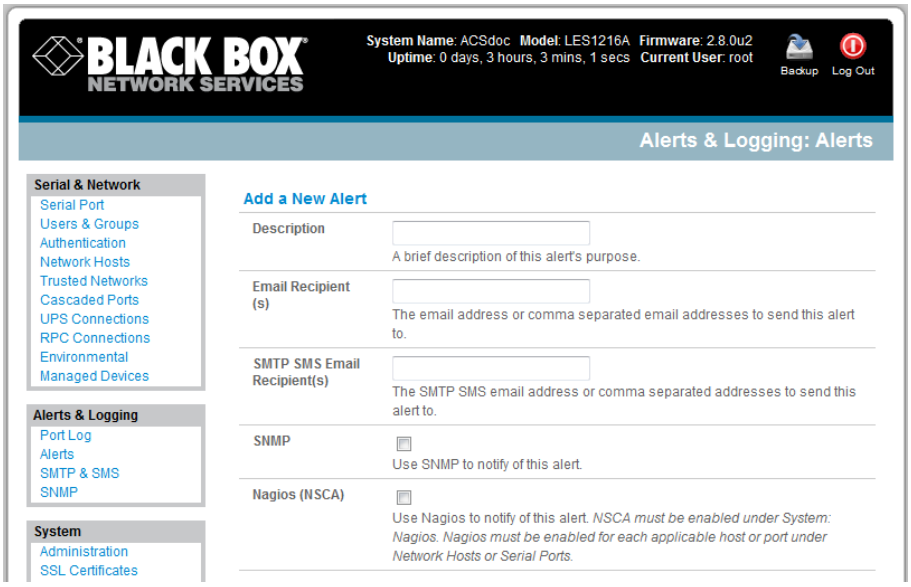
The Alert facility monitors the status of the *console server* and connected devices. When an alert event is triggered, the Alert facility notifies a nominated email address or SMS gateway, or the configured SNMP or Nagios server. The data stream from nominated serial ports can be monitored for matched patterns or flow control status changes can be configured to trigger alerts, as can user connections to serial ports and Hosts, or power events.



- Select **Alerts & Logging: Alerts,** which will display all the alerts currently configured. Click **Add Alert.**

7.2.1 Add a new alert

The first step is to specify the alert service that this event will use for sending notification, who to notify there, and what port/host/device is to be monitored:

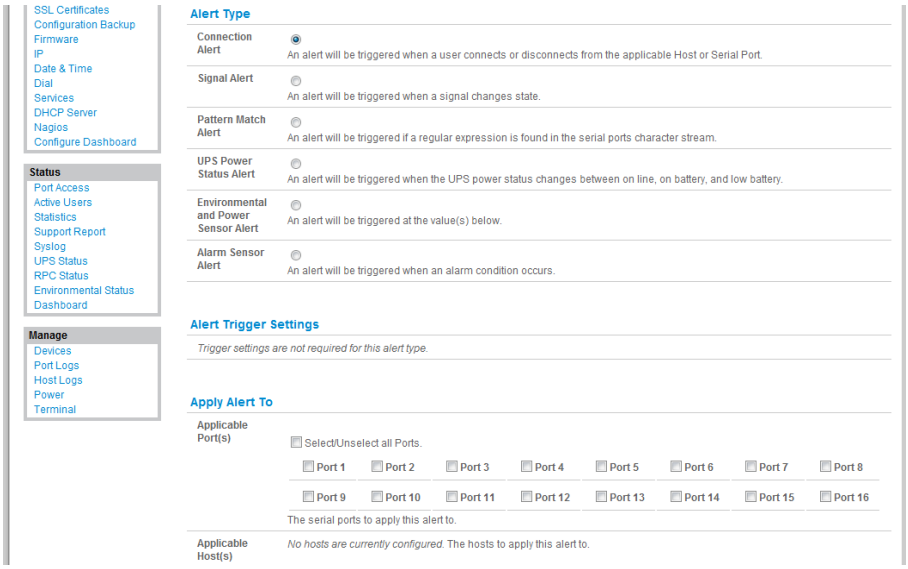

BLACK BOX
NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 3 hours, 3 mins, 1 secs Current User: root
Backup
Log Out
Alerts & Logging: Alerts
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Environmental
Managed Devices
Alerts & Logging
Port Log
Alerts
SMTP & SMS
SNMP
System
Administration
SSL Certificates
Add a New Alert
Description
A brief description of this alert's purpose.
Email Recipient (s)
The email address or comma separated email addresses to send this alert to.
SMTP SMS Email Recipient(s)
The SMTP SMS email address or comma separated addresses to send this alert to.
SNMP
Use SNMP to notify of this alert.
Nagios (NSCA)
Use Nagios to notify of this alert. NSCA must be enabled under System: Nagios. Nagios must be enabled for each applicable host or port under Network Hosts or Serial Ports.

- At **Add a New Alert,** enter a **Description** for this new alert.
- Nominate the email address for the **Email Recipient(s)** and/or the **SMS Recipient(s)** to be notified of the alert. For multiple recipients, enter comma separated addresses.
- Activate **SNMP** notification if an SNMP trap is to be sent for this event.
- Activate **Nagios** notification to use it for this event. In a SDT Nagios centrally managed environment, you can check the Nagios alert option. On the trigger condition (for matched patterns, logins, power events, and signal changes), an NSCA check “warning” result will be sent to the central Nagios server. This condition is displayed on the Nagios status screen and triggers a notification, which can cause the Nagios central server itself to send out an email or an SMS, page, etc.

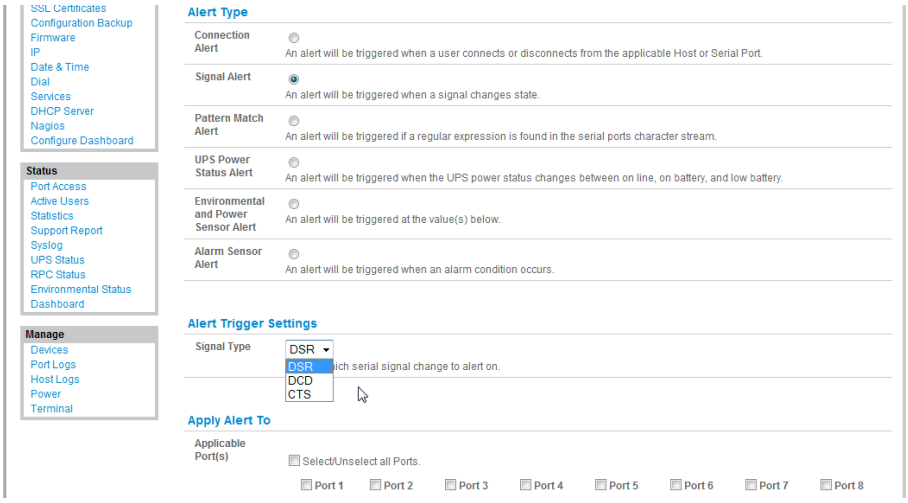
7.2.2 Configuring general alert types

Next, you must select the Alert Type (**Connection**, **Signal**, **Pattern Match**, **UPS Power Status**, **Environment and Power Sensor** or **Alarm Sensor**) to monitor. You can configure a selection of different Alert types and any number of specific triggers.

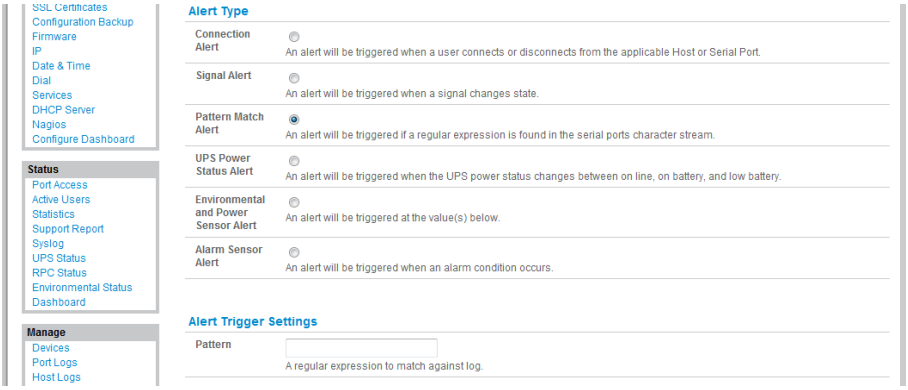
- **Connection Alert—**This alert will be triggered when a user connects or disconnects from the applicable Host or Serial Port, or when a Slave connects or disconnects from the applicable UPS (and you must specify the applicable connections to **Apply Alert To**).


SSL Certificates
Configuration Backup
Firmware
IP
Date & Time
Dial
Services
DHCP Server
Nagios
Configure Dashboard
Status
Port Access
Active Users
Statistics
Support Report
Syslog
UPS Status
RPC Status
Environmental Status
Dashboard
Manage
Devices
Port Logs
Host Logs
Power
Terminal
Alert Type
Connection Alert
An alert will be triggered when a user connects or disconnects from the applicable Host or Serial Port.
Signal Alert
An alert will be triggered when a signal changes state.
Pattern Match Alert
An alert will be triggered if a regular expression is found in the serial ports character stream.
UPS Power Status Alert
An alert will be triggered when the UPS power status changes between on line, on battery, and low battery.
Environmental and Power Sensor Alert
An alert will be triggered at the value(s) below.
Alarm Sensor Alert
An alert will be triggered when an alarm condition occurs.
Alert Trigger Settings
Trigger settings are not required for this alert type.
Apply Alert To
Applicable Port(s)
Select/Unselect all Ports.
Port 1 Port 2 Port 3 Port 4 Port 5 Port 6 Port 7 Port 8
Port 9 Port 10 Port 11 Port 12 Port 13 Port 14 Port 15 Port 16
The serial ports to apply this alert to.
Applicable Host(s)
No hosts are currently configured. The hosts to apply this alert to.

- **Serial Port Signal Alert**—This alert will be triggered when the specified signal changes state and applies to serial ports only. You must specify the particular **Signal Type** (DSR, DCD or CTS) trigger condition and the **Applicable Ports(s).**



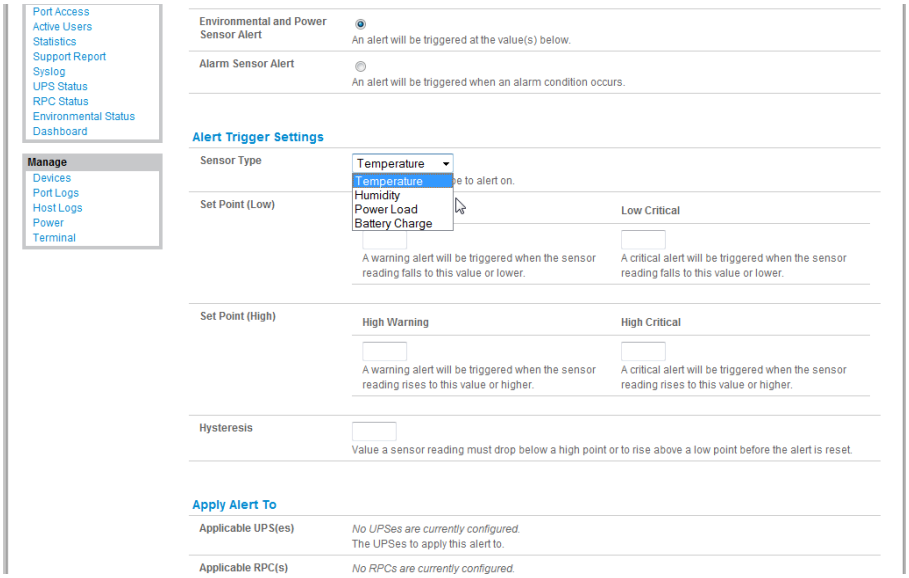
- **Serial Port Pattern Match Alert**—This alert will be triggered if a regular expression is found in the serial ports character stream that matches the regular expression you enter in the **Pattern** field. This alert type will only be applied to serial ports selected as **Applicable Ports(s).**



- **UPS Power Status Alert**— This alert will be triggered when the UPS power status changes between on line, on battery, and low battery. This status will only be monitored on the **Applicable UPS(es)** you select.
- **Environment and Power Alert**—(next section).
- **Alarm Sensor Alert**—(next section).

7.2.3 Configuring environment and power alert type

This alert type monitors UPSes, RPCs, power devices, and EMD environmental devices.


Port Access
Active Users
Statistics
Support Report
Syslog
UPS Status
RPC Status
Environmental Status
Dashboard
Manage
Devices
Port Logs
Host Logs
Power
Terminal
Environmental and Power Sensor Alert
An alert will be triggered at the value(s) below.
Alarm Sensor Alert
An alert will be triggered when an alarm condition occurs.
Alert Trigger Settings
Sensor Type
Temperature
Temperature
Humidity
Power Load
Battery Charge
Set Point (Low)
Low Critical
A warning alert will be triggered when the sensor reading falls to this value or lower.
A critical alert will be triggered when the sensor reading falls to this value or lower.
Set Point (High)
High Warning
High Critical
A warning alert will be triggered when the sensor reading rises to this value or higher.
A critical alert will be triggered when the sensor reading rises to this value or higher.
Hysteresis
Value a sensor reading must drop below a high point or to rise above a low point before the alert is reset.
Apply Alert To
Applicable UPS(es)
No UPSes are currently configured.
The UPSes to apply this alert to.
Applicable RPC(s)
No RPCs are currently configured.

- Select **Environment and Power Alert** to activate.
- Specify which **Sensor Type** to alert on (Temperature, Humidity, Power Load and Battery Charge).
- Set the levels at which **Critical** and/or **Warning** alerts are to be sent. You can also specify **High** and/or **Low Set Points** for sending alerts and the **Hysteresis** to be applied before resetting off the alerts.

Note Specify the **Set Point** values are in Degrees Centigrade for Temperature, Amps (Current) for Power Load, and % (Percentage) for both Humidity and Battery Charge.

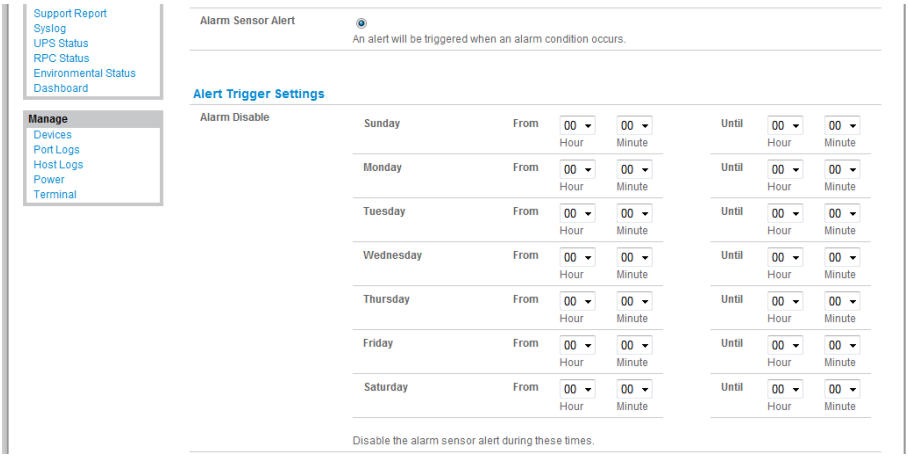
- Specify the applicable UPSes, RPCs (and RPC outlets), and Environmental Sensors to **Apply Alert To**.

Note An alert notification (SNMP, SMTP etc) is only sent out when there is a transition to or from a trigger event/level. For example, if a High temperature alert is set at 40 degrees with a 5 degree hysteresis then an High alert notification will be sent when the sensor temperature reads 40 degrees. The next alert will be sent when the temperature falls below 35 degrees. If the temp was over 40 degrees when the alert was first set, no high temp notification will be sent.

7.2.4 Configuring alarm sensor alert type

You can set an alert on sensor devices that may be attached to any EMD devices connected to the *console server*:

- Select **Alarm Sensor Alert** and then set the time windows when these sensors will not be monitored. For example, for a door open sensor, you may not want to deactivate the sensor alert monitoring during the working day (and the default 00:00 settings actively monitor the sensors 24/7).

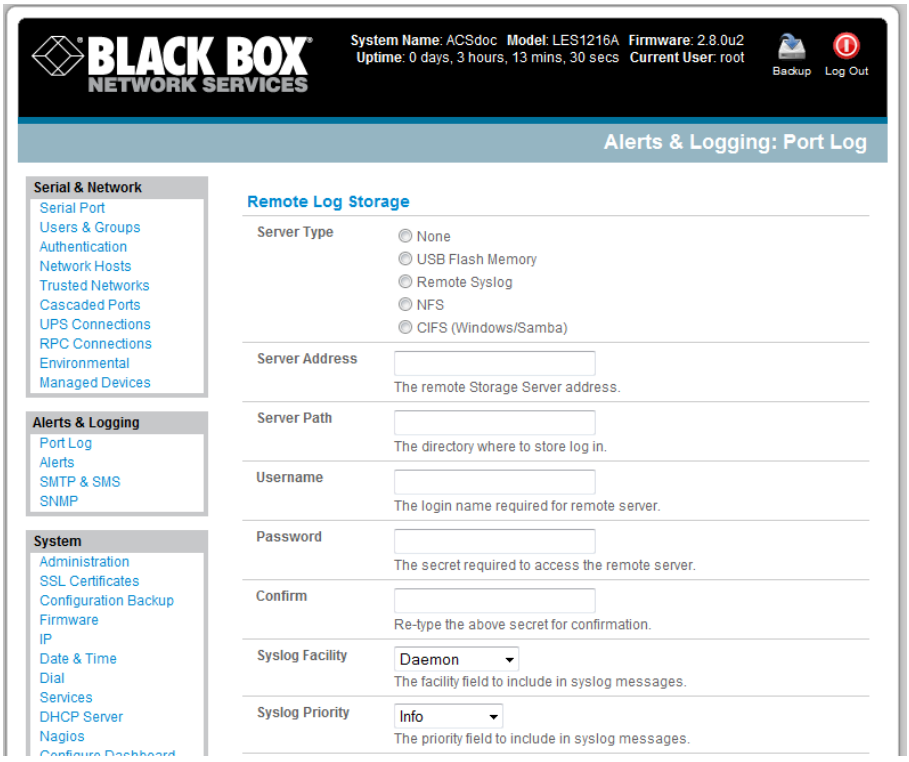


- Select the **Applicable Alarm Sensor(s)** for this alert and click **Apply.**

7.3 Remote Log Storage

Before activating Serial or Network Port Logging on any port or UPS logging, you must specify where those logs are to be saved:

- Select the **Alerts & Logging: Port Log** menu option and specify the **Server Type** to use, and the details to enable log server access.



7.4 Serial Port Logging

In *Console Server* mode, activity logs of all serial port activity can be maintained. These records are stored on an off-server, or in the Advanced Console Server flash memory. To specify which serial ports have activities recorded and to what level data is to be logged:

- Select **Serial & Network: Serial Port** and **Edit** the port to be logged.
- Specify the **Logging Level** of for each port as:

  Level 0 Turns off logging for the selected port.
  Level 1 Logs all connection events to the port.
  Level 2 Logs all data transferred to and from the port, all changes in hardware flow control status, and all *User* connection events.

- Click **Apply**

Note A cache of the most recent 8K of logged data per serial port is maintained locally (in addition to the Logs that are transmitted for remote/USB flash storage). To view the local cache of logged serial port data, select **Manage: Port Logs.**

7.5 Network TCP or UDP Port Logging

The LES1208A, LES1216A, and LES1248A models support optional logging of access to and communications with network attached Hosts.

- For each Host, when you set up the Permitted Services that you authorize to use, you also must set up the level of logging to maintain for each service.
- Specify the logging level to maintain for that particular TDC/UDP port/service, on that particular Host:

  Level 0 Turns off logging for the selected TDC/UDP port to the selected Host.
  Level 1 Logs all connection events to the port.
  Level 2 Logs all data transferred to and from the port.

- Click **Add** then click **Apply.**

Chapter 8 Power & Environmental Management

Introduction

Black Box *console server*s manage embedded software that you can use to manage connected Power Distribution Systems (PDUs), IPMI devices, and Uninterruptible Power Supplies (UPSs) supplied by a number of vendors, and some environmental monitoring devices.

8.1 Remote Power Control (RPC)

The *console server* Management Console monitors and controls Remote Power Control (RPC) devices using the embedded PowerMan and Network UPS Tools open source management tools and the Black Box power management software. RPCs include power distribution units (PDUs) and IPMI power devices.

You can control serial PDUs invariably using their command line console, so you could manage the PDU through the *console server* using a remote Telnet client. Also, you could use proprietary software tools supplied by the vendor. This generally runs on a remote Windows PC, and you could configure the *console server* serial port to operate with a serial COM port redirector in the PC (as detailed in *Chapter 4).*

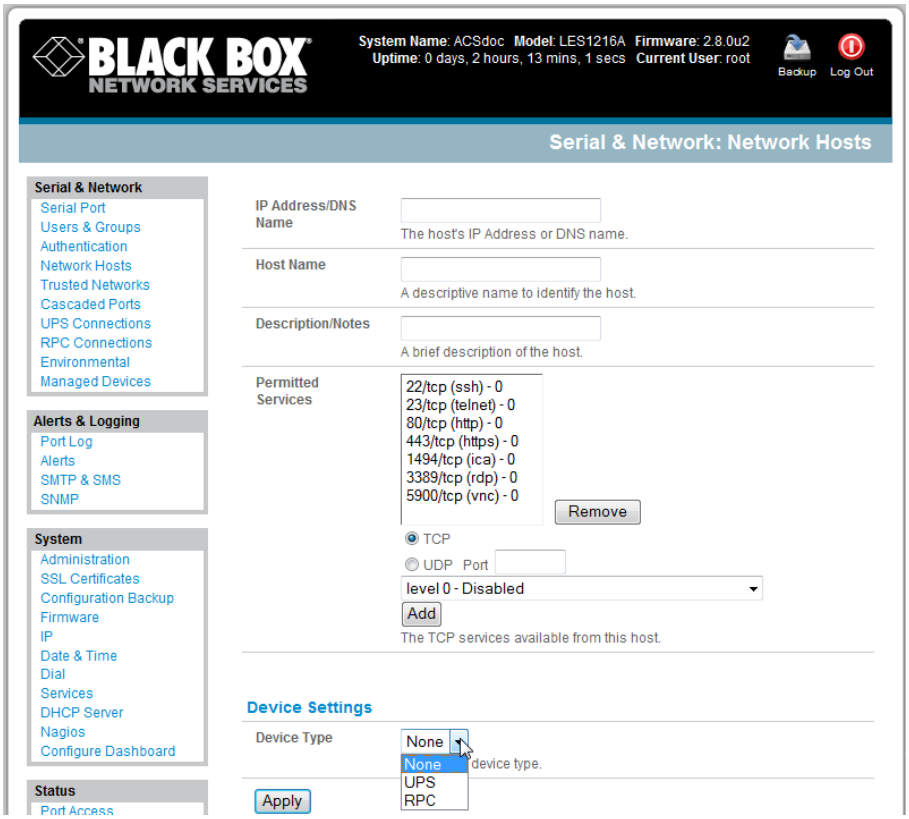
Similarly, you can control network-attached PDUs with a browser (for example, with SDT as detailed in *Chapter 6.3)*, an SNMP management package, or using the vendor-supplied control software. Servers and network-attached appliances with embedded IPMI service processors or BMCs invariably have their own management tools (like SoL) that provide secure management when connected with SDT Connector.

For simplicity, you can now control all these devices through one window using the Management Console's RPC remote power control tools.

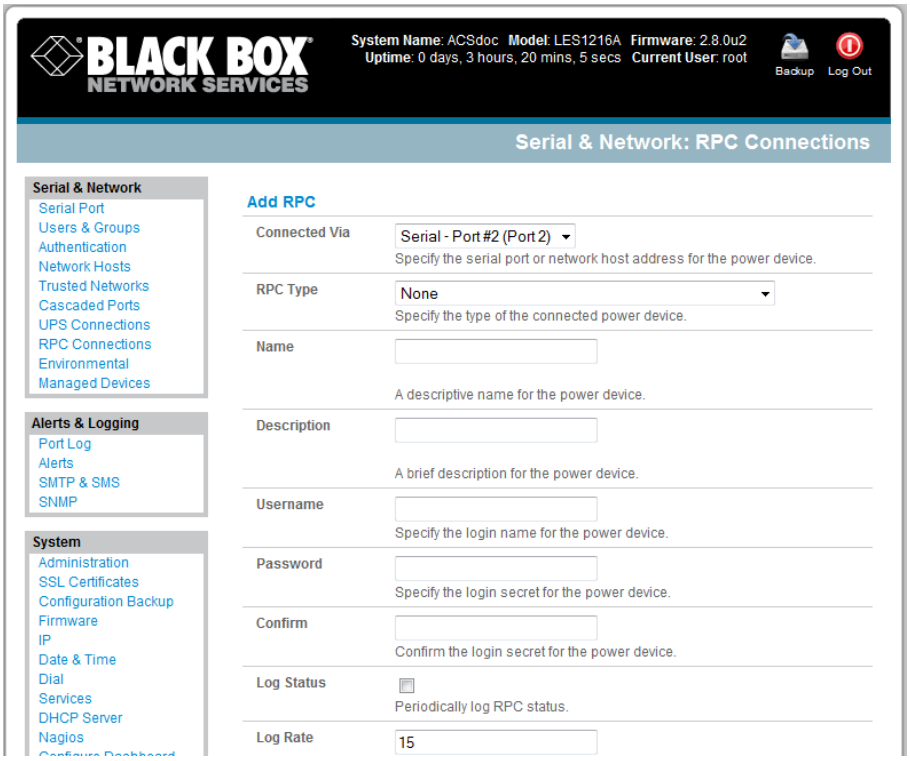
8.1.1 RPC connection

Serial and network connected RPCs must first be connected to, and configured to communicate with, the *console server*:

- For serial RPCs, connect the PDU to the selected serial port on the *console server*. From the **Serial and Network: Serial Port** menu, configure the **Common Settings** of that port with the RS-232 properties etc required by the PDU (refer to *Chapter 4.1.1 Common Settings*). Then select **RPC** as the **Device Type**.
- For each network-connected RPC, go to **Serial & Network: Network Hosts** menu and configure the RPC as a connected Host by specifying it as **Device Type: RPC** and clicking **Apply** (refer to *Section 4.4, Network Hosts*).



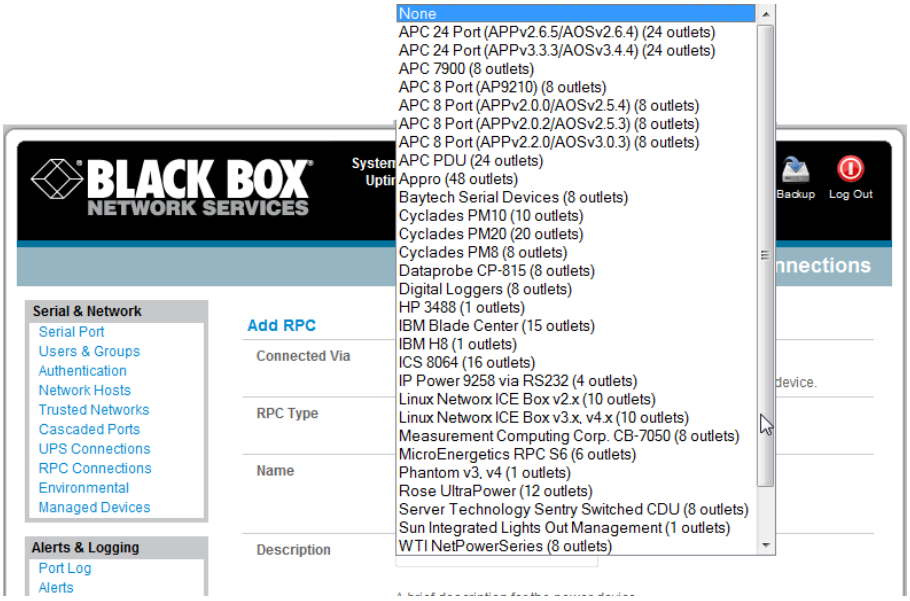
- Select the **Serial & Network: RPC Connections** menu. This will display all the RPC connections that have already been configured.
- Click **Add RPC**.
- **Connected Via** presents a list of serial ports and network Host connections that you have set up with device type RPC (but have yet to connect to a specific RPC device):
  - When you select **Connect Via** for a Network RPC connection, then the corresponding Host Name/Description that you set up for that connection will be entered as the **Name** and **Description** for the power device.
  - Or, if you select to **Connect Via** a Serial connection, enter a **Name** and **Description** for the power device.



➢ Select the appropriate **RPC Type** for the PDU (or IPMI) being connected:

- If you are connecting to the RPC via the network, you will be presented with the IPMI protocol options and the SNMP RPC Types currently supported by the embedded Network UPS Tools.

- If you are connecting to the RPC by a serial port, you will be presented with all the serial RPC types currently supported by the embedded PowerMan and the Black Box power manager:


None
APC 24 Port (APPv2.6.5/AOSv2.6.4) (24 outlets)
APC 24 Port (APPv3.3.3/AOSv3.4.4) (24 outlets)
APC 7900 (8 outlets)
APC 8 Port (AP9210) (8 outlets)
APC 8 Port (APPv2.0.0/AOSv2.5.4) (8 outlets)
APC 8 Port (APPv2.0.2/AOSv2.5.3) (8 outlets)
APC 8 Port (APPv2.2.0/AOSv3.0.3) (8 outlets)
APC PDU (24 outlets)
Appro (48 outlets)
Baytech Serial Devices (8 outlets)
Cyclades PM10 (10 outlets)
Cyclades PM20 (20 outlets)
Cyclades PM8 (8 outlets)
Dataprobe CP-815 (8 outlets)
Digital Loggers (8 outlets)
HP 3488 (1 outlets)
IBM Blade Center (15 outlets)
IBM H8 (1 outlets)
ICS 8064 (16 outlets)
IP Power 9258 via RS232 (4 outlets)
Linux Networx ICE Box v2.x (10 outlets)
Linux Networx ICE Box v3.x, v4.x (10 outlets)
Measurement Computing Corp. CB-7050 (8 outlets)
MicroEnergetics RPC S6 (6 outlets)
Phantom v3, v4 (1 outlets)
Rose UltraPower (12 outlets)
Server Technology Sentry Switched CDU (8 outlets)
Sun Integrated Lights Out Management (1 outlets)
WTI NetPowerSeries (8 outlets)

- Enter the **Username** and **Password** used to login into the RPC (Note that these login credentials are not related to the *Users* and access privileges you configured in *Serial & Networks: Users & Groups*).
- If you selected SNMP protocol, enter the SNMP v1 or v2c Community for Read/Write access (by default this would be "private").
- Check **Log Status** and specify the **Log Rate** (minutes between samples) if you want the status from this RPC to be logged. View these logs from the **Status: RPC Status** screen.
- Click **Apply.**
- For SNMP PDUs, the *console server* probes the configured RPC to confirm the RPC Type matches and reports the number of outlets it finds that can be controlled. If unsuccessful, it will report **Unable to probe outlets** and you'll need to check the RPC settings or network/serial connection.
- For serially connected RPC devices, a new Managed Device (with the same name as given to the RPC) will be created. The *console server* will then configure the RPC with the number of outlets specified in the selected RPC Type or will query the RPC itself for this information.

Note The Black Box *console server*s support most popular network and serial PDUs. If your PDU is not on the default list, then you can add support directly (as covered in Chapter 14—Advanced Configurations) or add the PDU support to either the Network UPS Tools or PowerMan open source projects.

Configure IPMI service processors and BMCs so that all authorized users can use the Management Console to remotely cycle power and reboot computers, even when their operating system is unresponsive. To set up IPMI power control, the *Administrator* first enters the IP address/domain name of the BMC or service processor (for example, a Dell DRAC) in **Serial & Network: Network Hosts,** then in **Serial & Network: RPC Connections** specifies the **RPC Type** to be IPMI1.5 or 2.0.

8.1.2 RPC access privileges and alerts

You can now set PDU and IPMI alerts using **Alerts & Logging: Alerts** (refer to *Chapter 7*). You can also assign which user can access and control which particular outlet on each RPC using **Serial & Network: User & Groups** (refer *Chapter 4*).

8.1.3 User power management

The Power Manager enables both *Users* and *Administrator*s to access and control the configured serial and network attached PDU power strips, and servers with embedded IPMI service processors or BMCs.

- Select the **Manage: Power** and the particular **Target** power device to be controlled (and the Outlet to be controlled if the RPC supports outlet level control).
- The outlet status is displayed and you can initiate the **Action** you want to take by selecting the appropriate icon:

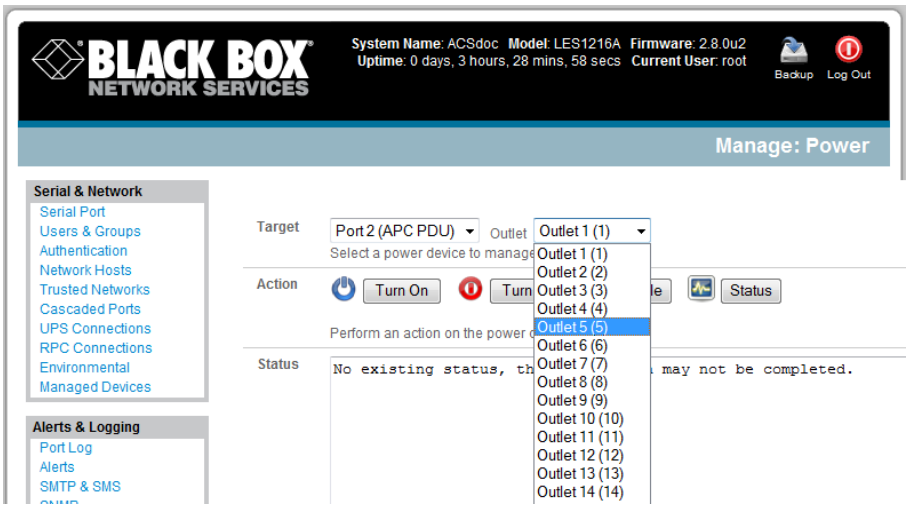
Turn ON

Turn OFF

Cycle

Status

You will only be presented with icons for those operations that are supported by the **Target** you have selected.



8.1.4 RPC status

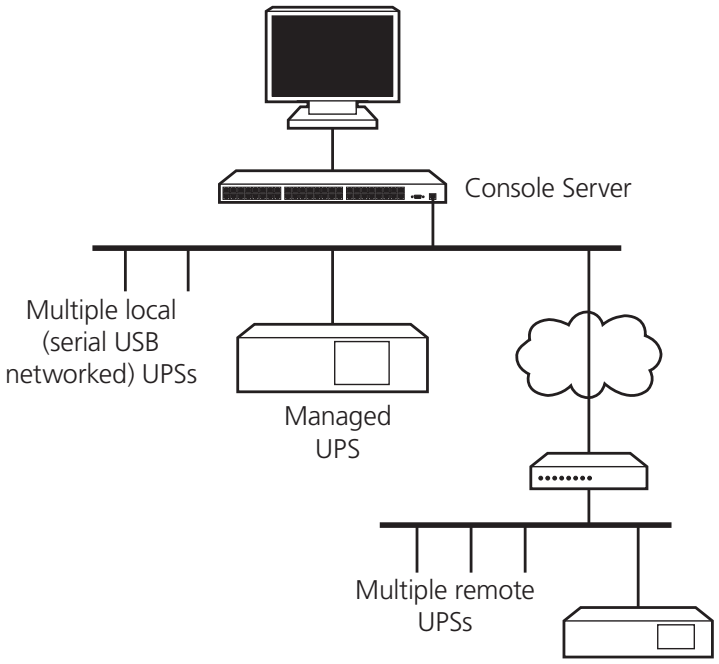
You can monitor the current status of your network and serially connected PDUs and IPMI RPCs.

- Select the **Status: RPC Status** menu and a table with the summary status of all connected RPC hardware will be displayed.
- Click on **View Log** or select the **RPCLogs** menu and you will be presented with a table of the history and detailed graphical information on the selected RPC.
- Click **Manage** to query or control the individual power outlet. This will take you to the **Manage: Power** screen.

8.2 Uninterruptible Power Supply Control (UPS)

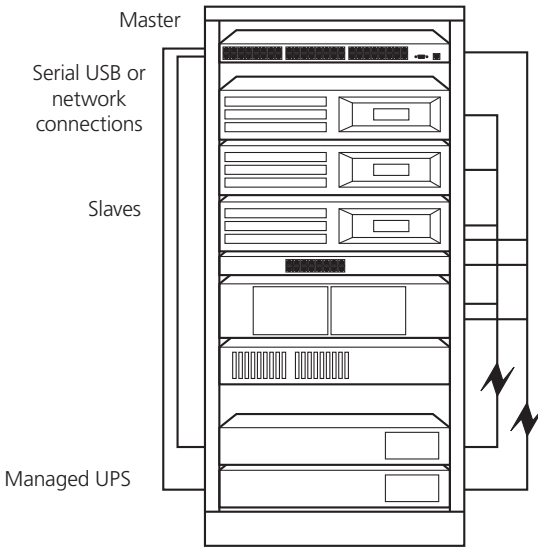
You can configure all Black Box *console servers* to manage locally and remotely connected UPS hardware using Network UPS Tools.

Network UPS Tools (NUT) is a group of open source programs that provide a common interface for monitoring and administering UPS hardware. These programs ensure safe shutdowns of the systems that are connected. NUT is built on a networked model with a layered scheme of drivers, server, and clients (covered in some detail in *Chapter 8.2.6*).



8.2.1 Managed UPS connections

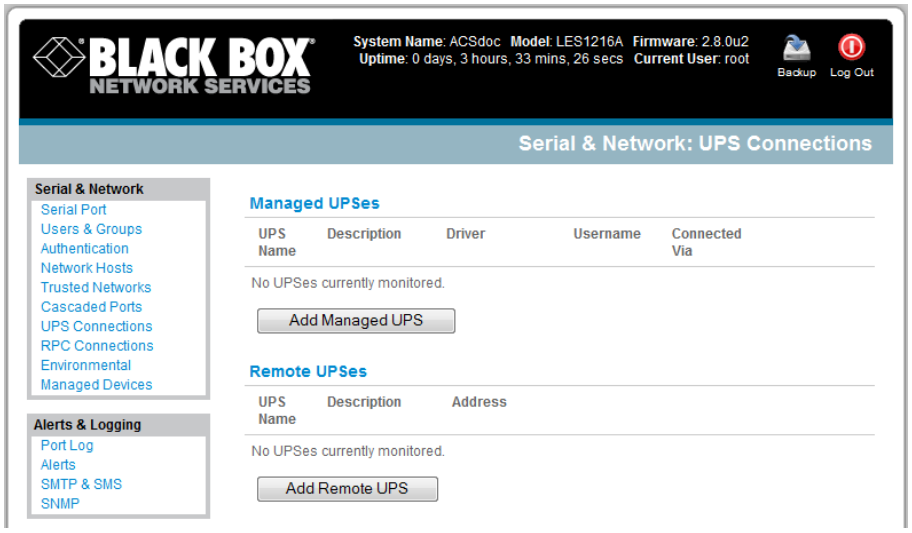
A ***Managed UPS*** is a UPS that is directly connected as a Managed Device to the *console server*. You can connect it via serial or USB cable or by the network. The *console server* becomes the *master* of this UPS, and runs a *upsd* server to allow other computers that are drawing power through the UPS (*slaves*) to monitor the UPS status and take appropriate action, such as shutdown when the UPS battery is low.



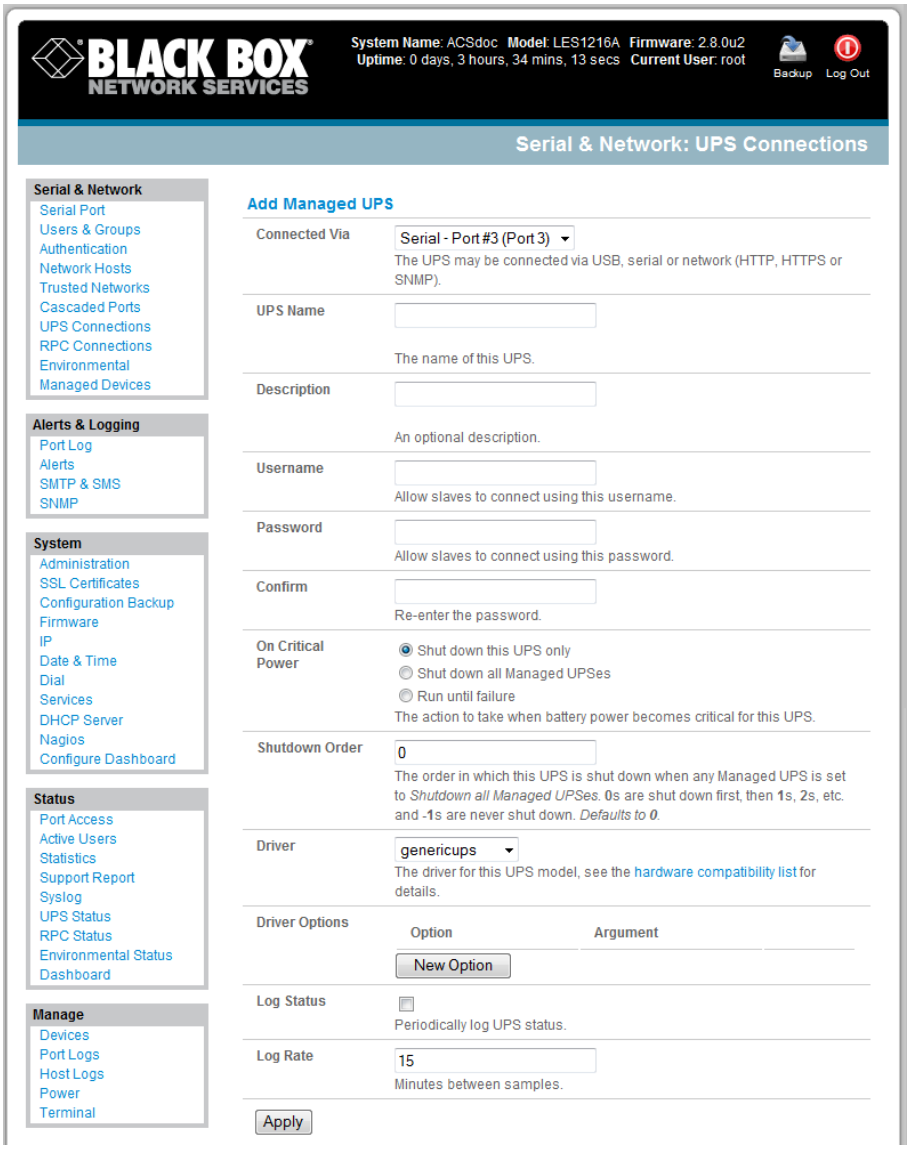
The *console server* may or may not be drawing power itself through the Managed UPS. When the UPS's battery power reaches critical, the *console server* signals and waits for *slaves* to shut down, then powers off the UPS.

Serial and network connected UPSes must first be connected to, and configured to communicate with the *console server*:

- For serial UPSes attach the UPS to the selected serial port on the *console server*. From the **Serial and Network: Serial Port** menu, configure the **Common Settings** of that port with the RS-232 properties, etc. required by the UPS (refer to *Chapter 4.1.1—Common Settings*). Then select **UPS** as the **Device Type**.
- For each network connected UPS, go to the **Serial & Network: Network Hosts** menu and configure the UPS as a connected Host by specifying it as **Device Type: UPS** and clicking **Apply.**
- No such configuration is required for USB connected UPS hardware.



- Select the **Serial & Network: UPS Connections** menu. The **Managed UPSes** section will display all the UPS connections that have already been configured.
- Click **Add Managed UPS**.



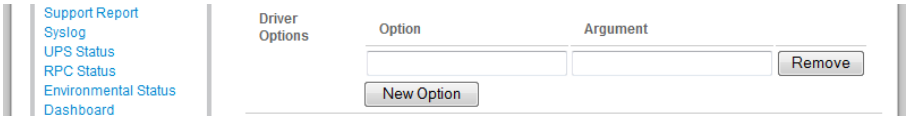
- Select if the UPS will be **Connected Via** USB, over a pre-configured serial port, or via SNMP/HTTP/HTTPS over the preconfigured network Host connection.
- When you select a network UPS connection, then the corresponding Host Name/Description that you set up for that connection will be entered as the **Name** and **Description** for the power device. Or, if you selected to **Connect Via** a USB or serial connection then you will need to enter a **Name** and **Description** for the power device (and these details will also be used to create a new Managed Device entry for the serial/USB connected UPS devices).
- Enter the login details. This **Username** and **Password** is used by *slaves* of this UPS (that is, other computers that are drawing power through this UPS) to connect to the *console server* to monitor the UPS status so they can shut themselves down when battery power is low. Monitoring will typically be performed using the *upsmon* client running on the slave server (refer to *Section 8.2.3*)

Note: These login credentials are not related to the *Users* and access privileges you configured in *Serial & Networks: Users & Groups.*

- If you have multiple UPSes and require them to be shut down in a specific order, specify the **Shutdown Order** for this UPS. This is a whole positive number, or *-1*. *0s* shut down first, then *1s*, *2s*, etc. *-1s* are not shut down at all. Defaults to *0.*

- Select the **Driver** that you will use to communicate with the UPS. Most *console servers* are preconfigured so the drop down menu presents a full selection of drivers from the latest Network UPS Tools (NUT version 2.4).

- Click **New Options** in **Driver Options** if you need to set driver-specific options for your selected NUT driver and hardware combination (more details at http://www.networkupstools.org/doc).



- Check **Log Status** and specify the **Log Rate** (minutes between samples) if you want the status from this UPS to be logged. You can view these logs from the **Status: UPS Status** screen.
- If you have enabled Nagios services, then you will presented with an option for Nagios monitoring. Check **Enable Nagios** to enable this UPS to be monitored using Nagios central management.
- Check **Enable Shutdown Script** if this is the UPS providing power to the *console server* itself and if a critical power failure occurs, you can perform any "*last gasp*" actions on the *console server* before power is lost. Place a custom script in */etc/config/scripts/ups-shutdown* (you may use the provided */etc/scripts/ups-shutdown* as a template). This script only runs when then UPS reaches critical battery status.
- Click **Apply.**

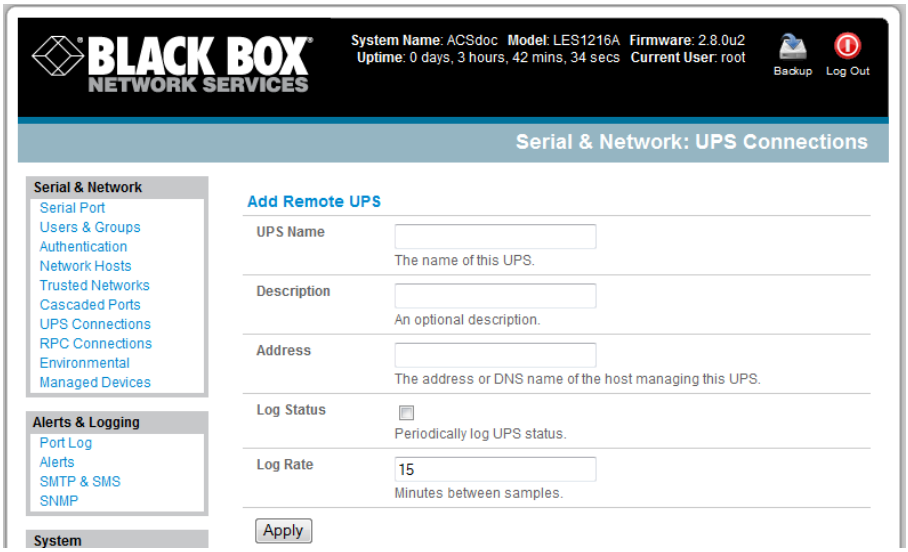
Note: You can also customize the *upsmon*, *upsd,* and *upsc* settings for this UPS hardware directly from the command line.

8.2.2 Remote UPS management

A ***Remote UPS*** is a UPS that is connected as a Managed Device to a remote *console server* that is monitored (but not managed) by your *console server*.

You can configure the *upsc* and *upslog* clients in the Black Box *console server* to monitor remote servers that are running Network UPS Tools managing their locally connected UPSes. These remote servers might be other Black Box *console servers* or generic Linux servers running NUT. You can centrally monitor all these distributed UPSes (which may be spread in a row in a data center, around a campus property, or across the country) through the one central *console server* window. To add a Remote UPS:

- Select the **Serial & Network: UPS Connections** menu. The **Remote UPSes** section will display all the remote UPS devices being monitored.
- Click **Add Remote UPS**.



- Enter the **Name** of the particular remote UPS that you want to remotely monitor. This name must be the name that the remote UPS was configured with on the remote *console server* (because the remote *console server* may itself have multiple UPSes attached that it manages locally with NUT). Optionally, enter a **Description.**
- Enter the IP **Address** or DNS name of the remote *console server** that is managing the remote UPS. (*This may be another Black Box *console server* or it may be a generic Linux server running Network UPS Tools.)

Note An example where centrally monitor remotely distributed UPSes is useful is a campus or large business site where there's a multitude of computer and other equipment sites spread afar, each with their own UPS supply … and many of these (particularly the smaller sites) will be USB or serially connected.

Having a *console server* at these remote sites would enable the system manager to centrally monitor the status of the power supplies at all sites, and centralize alarms. So he/she can be warned to initiate a call-out or shut-down.

- Check **Log Status** and specify the **Log Rate** (minutes between samples) if you want the status from this UPS to be logged. You can view these logs from the **Status: UPS Status** screen.
- Check **Enable Shutdown Script** if this remote UPS is the UPS providing power to the *console server* itself. If the UPS reaches critical battery status, the custom script in */etc/config/scripts/ups-shutdown* runs, enabling you to perform any "*last gasp*" actions.
- Click **Apply.**

8.2.3 Controlling UPS powered computers

One of the advantages of having a Managed UPS is that you can configure computers that draw power through that UPS to shut down gracefully if you have UPS problems.

For Linux computers, set up *upsmon* on each computer and direct them to monitor the *console server* that is managing their UPS. This will set the specific conditions that will be used to initiate a power down of the computer. Non-critical servers may be powered down some seconds after the UPS starts running

on battery. In contrast, more critical servers may not be shut down until a low battery warning is received). Refer to the online NUT documentation for details on how to do this:

http://eu1.networkupstools.org/doc/2.2.0/INSTALL.html

http://linux.die.net/man/5/upsmon.conf

http://linux.die.net/man/8/upsmon

An example upsmon.conf entry might look like:

MONITOR managedups@192.168.0.1 1 username password slave

- *managedups* is the UPS Name of the Managed UPS

- *192.168.0.1* is the IP address of the Black Box *console server*

- *1* indicates the server has a single power supply attached to this UPS

- *username* is the Username of the Managed UPS

- password is the Password of the Manager UPS

There are NUT monitoring clients available for Windows computers (WinNUT).

If you have an RPC (PDU), you can shut down UPS powered computers and other equipment if if the they don't have a client running (for example, communications, and surveillance gear). Set up a UPS alert and using this to trigger a script that controls a PDU to shut off the power (refer to *Chapter 15*).

8.2.4 UPS alerts

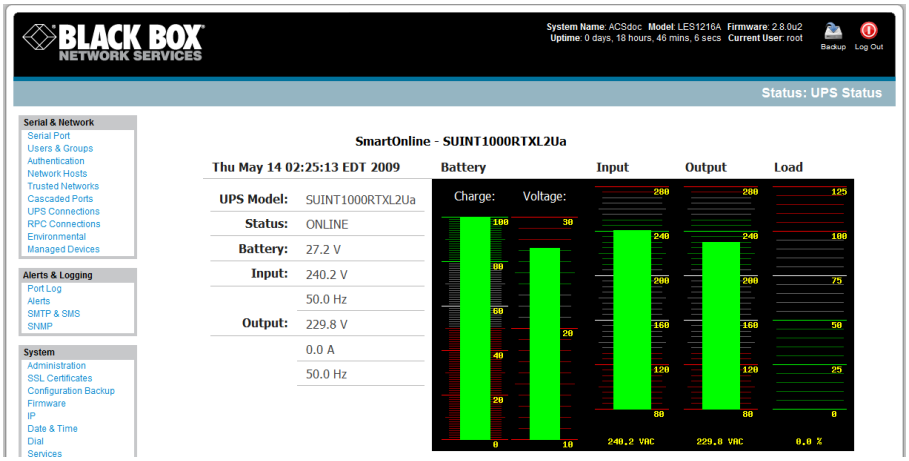
You can set UPS alerts using **Alerts & Logging: Alerts** (refer *Chapter — Alerts & Logging*).

8.2.5 UPS status

You can monitor the current status of your network, serially or USB connected Managed UPSes, and any configured Remote UPSes.

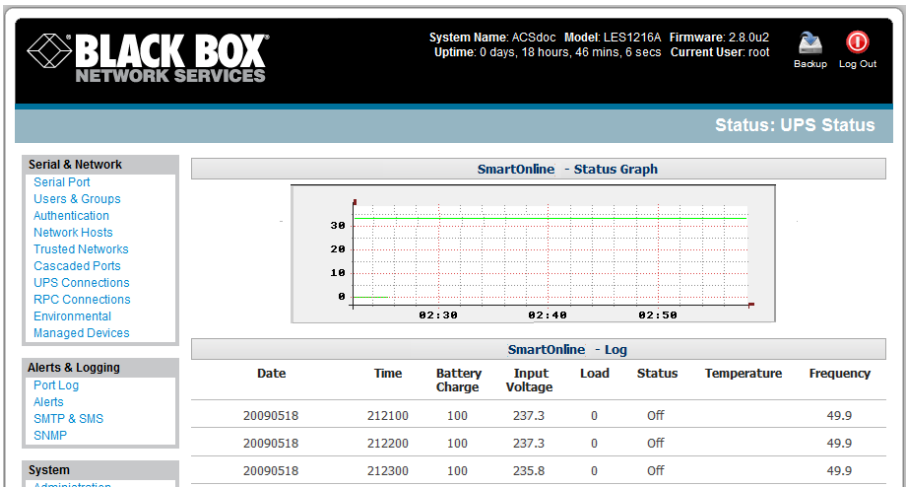
- Select the **Status: UPS Status** menu and a table with the summary status of all connected UPS hardware displays.

- Click on any particular UPS **System** name in the table and more detailed graphical information on the selected UPS System appears.



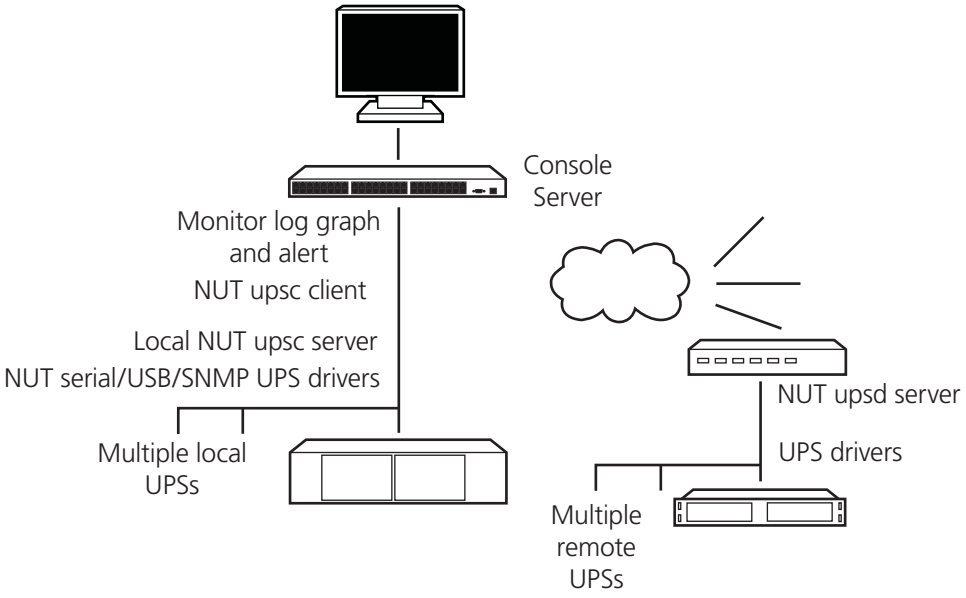
- Click on any particular **All Data** for any UPS System in the table for more status and configuration information about the selected UPS System.

- Select **UPS Logs** and you will be presented with the log table of the load, battery charge level, temperature, and other status information from all the Managed and Monitored UPS systems. This information will be logged for all UPSes that were configured with **Log Status** checked. The information is also presented graphically.



8.2.6 Overview of Network UPS Tools (NUT)

NUT is built on a networked model with a layered scheme of drivers, server and clients. Configure NUT using the Management Console as described above, or configure the tools and manage the UPSes directly from the command line. This section provides an overview of NUT. You can find full documentation at *http://www.networkupstools.org/doc*.



NUT is built on a networked model with a layered scheme of drivers, server and clients:

- The **driver** programs talk directly to the UPS equipment and run on the same host as the NUT network server (*upsd*). Drivers are provided for a wide assortment of equipment from most of the popular UPS vendors and understand the specific language of each UPS. They communicate with serial, USB, and SNMP network connected UPS hardware and map the communications back to a compatibility layer. This means both an expensive "smart" protocol UPS and a simple "power strip" model can be handled transparently.

- The NUT network **server** program *upsd* is responsible for passing status data from the drivers to the client programs via the network. *upsd* can cache the status from multiple UPSes and then serve this status data to many clients. *upsd* also contains access control features to limit the abilities of the clients (only authorized hosts may monitor or control the UPS hardware).

- There are a number of NUT **clients** that connect to *upsd* to check on the status of the UPS hardware and do things based on the status. These clients can run on the same host as the NUT server or they can communicate with the NUT server over the network (enabling them to monitor any UPS anywhere):

    - The *upsc* client provides a quick way to poll the status of a UPS server. Use it inside shell scripts and other programs that need UPS data but don't want to include the full interface.
    - The *upsmon* client enables servers that draw power through the UPS to shutdown gracefully when the battery power reaches critical.
    - There are also logging clients (*upslog*) and third party interface clients (Big Sister, Cacti, Nagios, Windows, and more. Refer *www.networkupstools.org/client-projects*.)

- The latest release of NUT (2.4) also controls PDU systems. It can do this either natively using SNMP or through a *binding* to Powerman (open source software from Livermore Labs that also is embedded in Black Box *console server*s).

These NUT clients and servers all are embedded in each Black Box *console server* (with a Management Console presentation layer added) —and they also are run remotely on distributed *console server*s and other remote NUT monitoring systems. This layered distributed NUT architecture enables:

- Multiple manufacturer support: NUT can monitor UPS models from 79 different manufacturers—and PDUs from a growing number of vendors—with a unified interface.
- Multiple architecture support: NUT can manage serial and USB connected UPS models with the same common interface. Network-connected USB and PDU equipment can also be monitored using SNMP.
- Multiple clients monitoring one UPS: Multiple systems may monitor a single UPS using only their network connections. There is a wide selection of client programs that support monitoring UPS hardware via NUT (Big Sister, Cacti, Nagios and more).
- Central management of multiple NUT servers: A central NUT client can monitor multiple NUT servers that may be distributed throughout the data center, across a campus, or around the world.

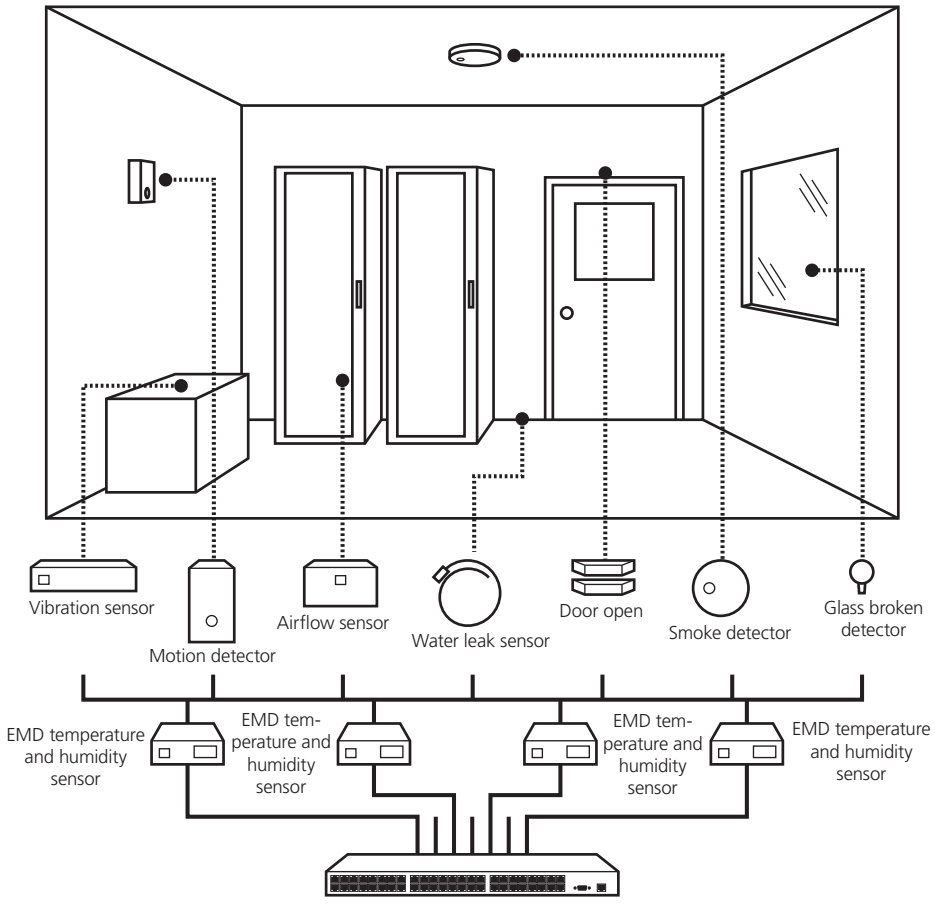
NUT supports the more complex power architectures found in data centers, communications centers, and distributed office environments where many UPSes from many vendors power many systems with

many clients. Each of the larger UPSes power multiple devices, and many of these devices are in turn dual powered.

8.3 Environmental Monitoring

The Environmental Monitor Device (EMD) connects to any Black Box *console server* serial port and each *console server* can support multiple EMDs. Each EMD device has one temperature and one humidity sensor and one or two general-purpose status sensors that you can connect to a smoke detector, water detector, vibration, or open-door sensor.

Using the Management Console, *Administrator*s can view the ambient temperature (in °C) and humidity (percentage), and set the EMD to automatically send alarms progressively from warning levels to critical alerts.



8.3.1 Connecting the EMD

The Environmental Monitor Device (EMD) connects to any serial port on the *console server* via a special EMD Adapter and standard CAT5 cable. The EMD is powered over this serial connection and communicates using a custom handshake protocol. It is not an RS-232 device and should not be connected without the adapter:

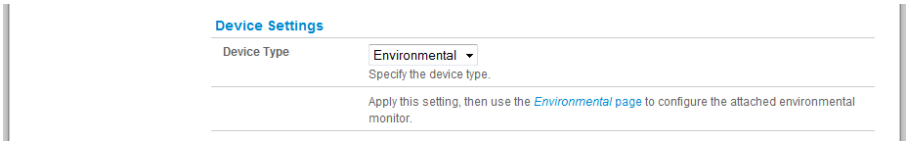
- Plug the male RJ plug on the EMD Adapter into EMD and then connect it to the *console server* serial port using the provided UTP cable. If the 6-foot (2-meter) UTP cable provided with the EMD is not long enough, you can replace it with a standard CAT5 UTP cable up to 33 feet (10 meters) long.

- Screw the bare wires on any smoke detector, water detector, vibration sensor, open-door sensor, or general purpose open/close status sensors into the terminals on the EMD.

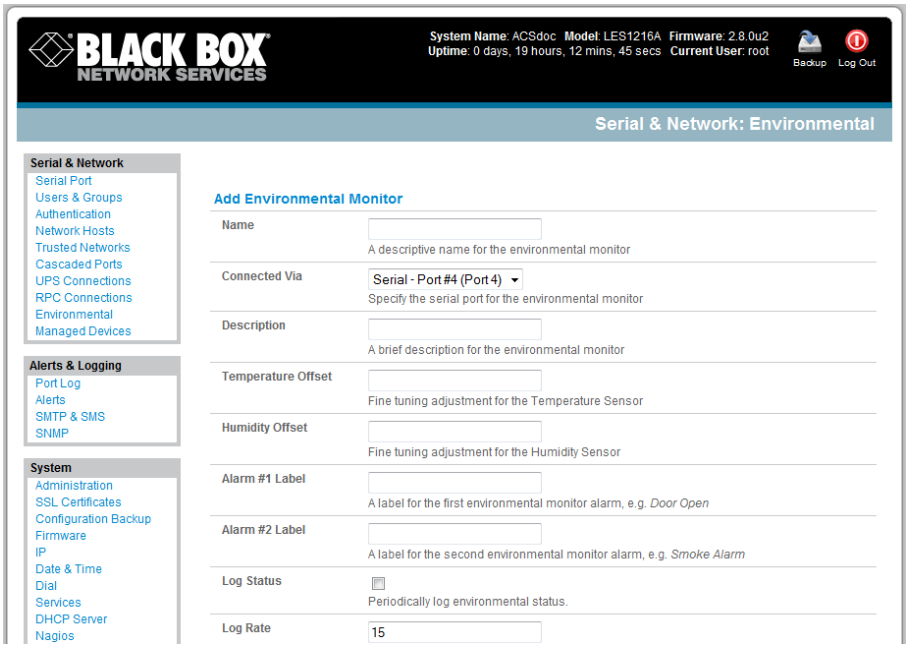
Note: You can attach two external sensors onto the terminals on EMDs that are connected to LES1108A, LES1116A, and LES1148A *console servers*. LES1208A, LES1216A, and LES1248A *console servers* only support attaching a single sensor to each EMD.

You can only use the EMD with a Black Box *console server*; you cannot connect it to standard RS-232 serial ports on other appliances.

- Select **Environmental** as the **Device Type** in the **Serial & Network: Serial Port** menu for the port to which the EMD will be attached. No particular Common Settings are required.


Device Settings
Device Type
Environmental
Specify the device type.
Apply this setting, then use the Environmental page to configure the attached environmental monitor.

- Click **Apply.**
- Select the **Serial & Network: Environmental** menu. This will display all the EMD connections that have already been configured.
- Click **Add.**



- Enter a **Name** and optionally a **Description** for the EMD and select the pre-configured serial port that the EMD will be **Connected Via.**
- You may optionally calibrate the EMD with a Temperature Offset (+ or - °C) or Humidity Offset (+ or percent).
- Provide **Labels** for each of the two alarms (if used).
- Check **Log Status** and specify the **Log Rate** (minutes between samples) if you want to log the status from this EMD. These logs can be views from the **Status: Environmental Status** screen.
- Click **Apply**. This will also create a new Managed Device (with the same name).

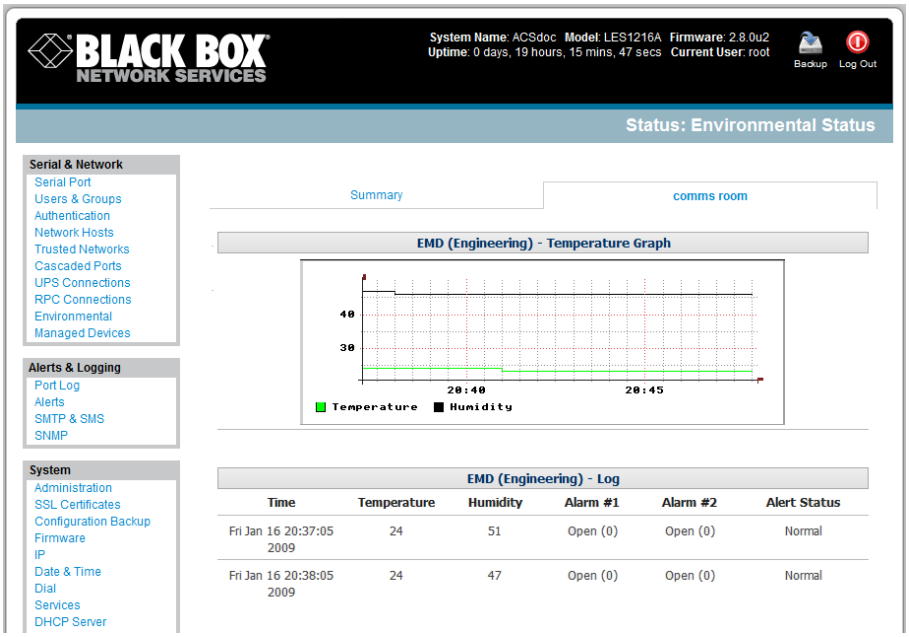
8.3.2 Environmental alerts

You can now set temperature, humidity and probe status alerts using **Alerts & Logging: Alerts** (refer to *Chapter 7*).

8.3.3 Environmental status

You can monitor the current status of all EMDs and their probes.

- Select the **Status: Environmental Status** menu and a table with the summary status of all connected EMD hardware will be displayed.
- Click on **View Log** or select the **Environmental Logs** menu and you will be presented with a table and graphical plot of the selected EMD's log history.


BLACK BOX
NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 19 hours, 15 mins, 47 secs Current User: root
Backup
Log Out
Status: Environmental Status
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Environmental
Managed Devices
Alerts & Logging
Port Log
Alerts
SMTP & SMS
SNMP
System
Administration
SSL Certificates
Configuration Backup
Firmware
IP
Date & Time
Dial
Services
DHCP Server
Summary
comms room
EMD (Engineering) - Temperature Graph
40
30
20:40
20:45
Temperature
Humidity
EMD (Engineering) - Log
Time
Temperature
Humidity
Alarm #1
Alarm #2
Alert Status
Fri Jan 16 20:37:05 2009
24
51
Open (0)
Open (0)
Normal
Fri Jan 16 20:38:05 2009
24
47
Open (0)
Open (0)
Normal

Chapter 9 Authentication

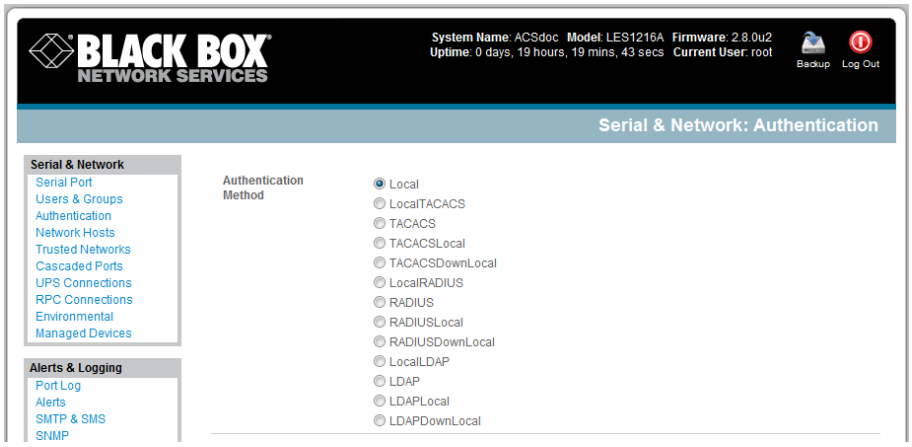
Introduction

The *console server* is a dedicated Linux computer with a myriad of popular and proven Linux software modules for networking, secure access (OpenSSH), and communications (OpenSSL), and sophisticated user authentication (PAM, RADIUS, TACACS+ and LDAP).

- This chapter details how the *Administrator* can use the Management Console to establish remote AAA authentication for all connections to the *console server* and attached serial and network host devices.
- This chapter also covers how to establish a secure link to the Management Console using HTTPS and using OpenSSL and OpenSSH to establish a secure Administration connection to the *console server.*

9.1 Authentication Configuration

Authentication can be performed locally, or remotely using an LDAP, Radius, or TACACS+ authentication server. The default authentication method for the *console server* is Local.



Any authentication method that is configured will be used for authentication of any user who attempts to log in through Telnet, SSH, or the Web Manager to the *console server* and any connected serial port or network host devices.

You can configure the *console server* to the default (**Local**) or using an alternate authentication method (**TACACS**, **RADIUS,** or **LDAP**). Optionally, you can select the order in which local and remote authentication is used:

Local *TACACS* */RADIUS/LDAP*: Tries local authentication first, falling back to remote if local fails.

*TACACS /RADIUS/****LDAP*** **Local**: Tries remote authentication first, falling back to local if remote fails.

*TACACS /RADIUS/**LDAP*** **Down Local**: Tries remote authentication first, falling back to local if the remote authentication returns an error condition (for example, if the remote authentication server is down or inaccessible).

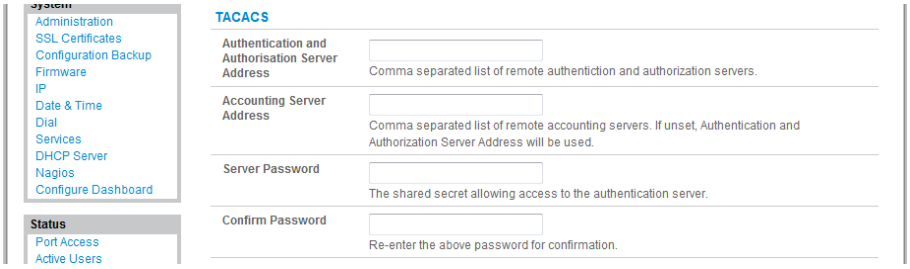
9.1.1 Local authentication

- Select **Serial and Network: Authentication** and check **Local**.
- Click **Apply.**

9.1.2 TACACS authentication

Perform the following procedure to configure the TACACS+ authentication method to use whenever the *console server* or any of its serial ports or hosts is accessed:

- Select **Serial and Network: Authentication** and check **TACAS** or **LocalTACACS** or **TACACSLocal** or **TACACSDownLocal**



- Enter the **Server Address** (IP or host name) of the remote Authentication/Authorization server. Multiple remote servers may be specified in a comma-separated list. Each server is tried in succession.
- In addition to multiple remote servers, you can also enter separate lists of Authentication/ Authorization servers and Accounting servers. If no Accounting servers are specified, the Authentication/Authorization servers are used instead.
- Enter the **Server Password.**
- Click **Apply.** TACAS+ remote authentication will now be used for all user access to *console server* and serially or network attached devices.

TACACS+ The Terminal Access Controller Access Control System (TACACS+) security protocol is a recent protocol developed by Cisco. It provides detailed accounting information and flexible administrative control over the authentication and authorization processes. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide authentication, authorization, and accounting services independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon. There is a draft RFC detailing this protocol. You can find further information on configuring remote TACACS+ servers at the following sites:

http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080094e99.shtml

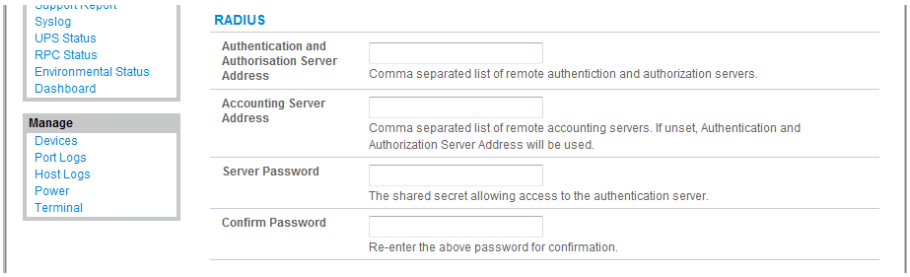
http://www.cisco.com/en/US/products/sw/secursw/ps4911/products_user_guide_chapter09186a00800eb6d6.html

http://cio.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt2/sctplus.htm

9.1.3 RADIUS authentication

Perform the following procedure to configure the RADIUS authentication method to use whenever the *console server* or any of its serial ports or hosts is accessed:

- Select **Serial and Network: Authentication** and check **RADIUS** or **LocalRADIUS** or **RADIUSLocal** or **RADIUSDownLocal**.



- Enter the **Server Address** (IP or host name) of the remote Authentication/ Authorization server. Multiple remote servers may be specified in a comma-separated list. Each server is tried in succession.
- In addition to multiple remote servers, you can also enter separate lists of Authentication/ Authorization servers and Accounting servers. If no Accounting servers are specified, the Authentication/Authorization servers are used instead.
- Enter the **Server Password.**
- Click **Apply.** RADIUS remote authentication will now be used for all user access to *console server* and serially or network-attached devices.

RADIUS The Remote Authentication Dial-In User Service (RADIUS) protocol was developed by Livingston Enterprises as an access server authentication and accounting protocol. The RADIUS server can support a variety of methods to authenticate a user. When it is provided with the username and original password given by the user, it can support PPP, PAP, or CHAP, UNIX login, and other authentication mechanisms. You can find further information on configuring remote RADIUS servers at the following sites:

http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/DepKit/d4fe8248-eecd-49e4-88f6-9e304f97fefc.mspx

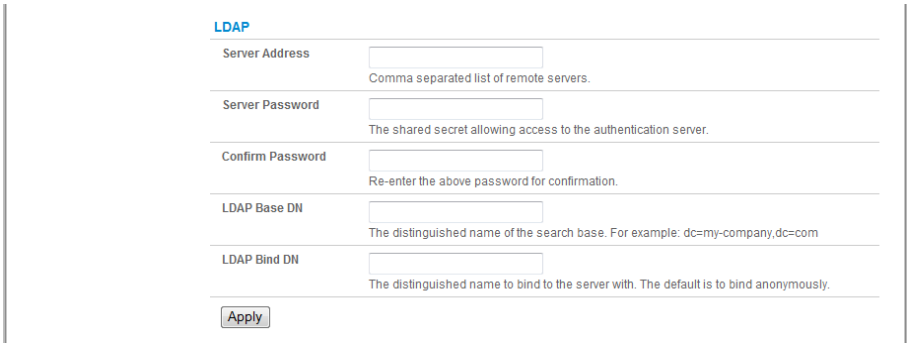
http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a00800945cc.shtml

http://www.freeradius.org/

9.1.4 LDAP authentication

Perform the following procedure to configure the LDAP authentication method to use whenever the *console server* or any of its serial ports or hosts is accessed:

- Select **Serial and Network: Authentication** and check **LDAP** or **LocalLDAP** or **LDAPLocal** or **LDAPDownLocal**


LDAP
Server Address
Comma separated list of remote servers.
Server Password
The shared secret allowing access to the authentication server.
Confirm Password
Re-enter the above password for confirmation.
LDAP Base DN
The distinguished name of the search base. For example: dc=my-company,dc=com
LDAP Bind DN
The distinguished name to bind to the server with. The default is to bind anonymously.
Apply

- Enter the **Server Address** (IP or host name) of the remote Authentication server. Multiple remote servers may be specified in a comma-separated list. Each server is tried in succession.
- Enter the **Server Password.**

Note To interact with LDAP requires that the user account exist on our *console server* to work with the remote server. (You can't just create the user on your LDAP server and not tell the *console server* about it.) You need to add the user account.

- Click **Apply.** LDAP remote authentication will now be used for all user access to *console server* and serially or network attached devices.

LDAP The Lightweight Directory Access Protocol (LDAP) is based on the X.500 standard, but is significantly simpler and more readily adapted to meet custom needs. The core LDAP specifications are all defined in RFCs. LDAP is a protocol used to access information stored in an LDAP server. You can find further information on configuring remote RADIUS servers at the following sites:

http://www.ldapman.org/articles/intro_to_ldap.html

http://www.ldapman.org/servers.html

http://www.linuxplanet.com/linuxplanet/tutorials/5050/1/

http://www.linuxplanet.com/linuxplanet/tutorials/5074/4/

9.1.5 RADIUS/TACACS User Configuration

Users may be added to the local *console server* appliance. If they are not added and they log in via remote AAA, a user will be added for them. This user will not show up in the Black Box configurators unless they are specifically added, at which point they are transformed into a completely local user. The newly added user must authenticate from the remote AAA server, and will have no access if it is down.

If a local user logs in, they may be authenticated/authorized from the remote AAA server, depending on the chosen priority of the remote AAA. A local user's authorization is the union of local and remote privileges.

Example 1:

> User Tim is locally added, and has access to ports 1 and 2. He is also defined on a remote TACACS server, which says he has access to ports 3 and 4. Tim may log in with either his local or TACACS password, and will have access to ports 1 through 4. If TACACS is down, he will need to use his local password, and will only be able to access ports 1 and 2.

Example 2:

> User Ben is only defined on the TACACS server, which says he has access to ports 5 and 6. When he attempts to log in, a new user will be created for him, and he will be able to access ports 5 and 6. If the TACACS server is down he will have no access.

Example 3:

> User Paul is defined on a RADIUS server only. He has access to all serial ports and network hosts.

Example 4:

> User Don is locally defined on an appliance using RADIUS for AAA. Even if Don is also defined on the RADIUS server, he will only have access to those serial ports and network hosts he has been authorized to use on the appliance.

If a "no local AAA" option is selected, then root will still be authenticated locally.

You can add remote users to the admin group via either RADIUS or TACACS. Users may have a set of authorizations set on the remote TACACS server. Users automatically added by RADIUS will have authorization for all resources, whereas those added locally will still need their authorizations specified.

LDAP has not been modified, and will still need locally defined users.

9.2 PAM (Pluggable Authentication Modules)

The *console server* supports RADIUS, TACACS+, and LDAP for two-factor authentication *via* PAM (Pluggable Authentication Modules). PAM is a flexible mechanism for authenticating users. Nowadays, a number of new ways of authenticating users have become popular. The challenge is that each time a new authentication scheme is developed, you need to rewrite all the necessary programs (login, ftpd, *etc*.) to support it.

PAM provides a way to develop programs that are independent of authentication scheme. These programs need "authentication modules" to be attached to them at run-time in order to work. Which authentication module is attached depends on the local system setup and is at the discretion of the local *Administrator*.

The *console server* family supports PAM with the following modules added for remote authentication:

RADIUS	- pam_radius_auth	(http://www.freeradius.org/pam_radius_auth/)
TACACS+	- pam_tacplus	(http://echelon.pl/pubs/pam_tacplus.html)
LDAP	- pam_ldap	(http://www.padl.com/OSS/pam_ldap.html)

Further modules can be added as required.

Changes may be made to files in /etc/config/pam.d/ that will persist, even if the authentication configurator runs.

- Users added on demand:

  When a user attempts to log in, but does not already have an account on the *console server*, a new user account will be created. This account will have no rights, and no password set. It will not appear in the Black Box configuration tools.

  Automatically added accounts will not be able to log in if the remote servers are unavailable. RADIUS users are currently assumed to have access to all resources, so they will only be authorized to log in to the *console server*. RADIUS users will be authorized each time they access a new resource.

- Admin rights granted over AAA:

  Users may be granted *Administrator* rights via networked AAA. For TACACS a priv-lvl of 12 of above indicates an *Administrator*. For RADIUS, *Administrator*s are indicated via the Framed Filter ID. (See the example configuration files below for example.)

- Authorization via TACACS for both serial ports and host access:

  Permission to access resources may be granted via TACACS by indicating a Black Box Appliance and a port or networked host the user may access. (See the example configuration files below for example.)

TACACS Example:

```
user = tim {
  service = raccess {
     priv-lvl = 11
     port1 = les1116/port02
     port2 = 192.168.254.145/port05
  }
  global = cleartext mit
}
```

RADIUS Example:

```
paul    Cleartext-Password := "luap"
        Service-Type = Framed-User,
        Fall-Through = No,
        Framed-Filter-Id=":group_name=admin"
```

The list of groups may include any number of entries separated by a comma. If the admin group is included, the user will be made an *Administrator*.

If there is already a Framed-Filter-Id, simply add the list of *group_names* after the existing entries, including the separating colon ":".

9.3 SSL Certificate

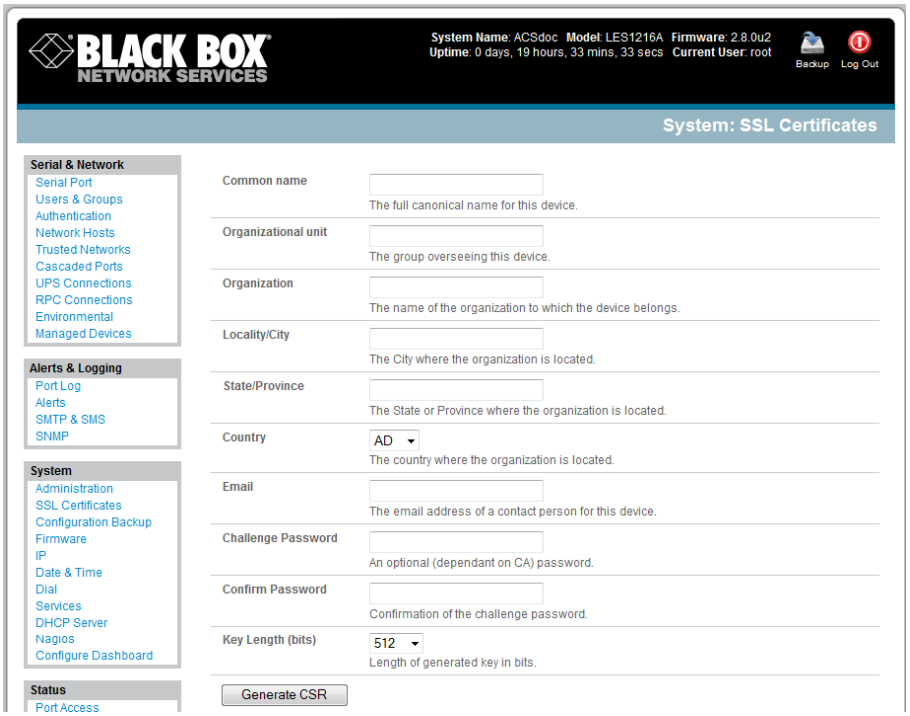
The *console server* uses the Secure Socket Layer (SSL) protocol for encrypted network traffic between itself and a connected user. When establishing the connection, the *console server* has to expose its identity to the user's browser using a cryptographic certificate. The default certificate that comes with the *console server* device upon delivery is for testing purposes only.

The System Administrator should not rely on the default certificate as the secured global access mechanism for use through Internet.

- Activate your preferred browser and enter https:// *IP address.* Your browser may respond with a message that verifies the security certificate is valid but notes that it is not necessarily verified by a certifying authority. To proceed, you need to click *yes* if you are using Internet Explorer or select *accept this certificate permanently* (or *temporarily*) if you are using Mozilla Firefox.
- You will then be prompted for the *Administrator* account and password as normal.

We recommend that you generate and install a new base64 X.509 certificate that is unique for a particular *console server*.

To do this, the *console server* must be enabled to generate a new cryptographic key and the associated Certificate Signing Request (CSR) that needs to be certified by a Certification Authority (CA). A certification authority verifies that you are the person who you claim you are, and signs and issues a SSL certificate to you. To create and install a SSL certificate for the *console server*:



- Select **System: SSL Certificate** and fill out the fields as explained below:

  Common name This is the network name of the *console server* once it is installed in the network (usually the fully qualified domain name). It is identical to the name that is used to access the *console server* with a web browser (without the "http://" prefix). In case the name given here and the actual network name differ, the browser will pop up a security warning when the *console server* is accessed using HTTPS.

  Organizational Unit Use this field to specify which department within an organization the *console server* belongs to.

  Organization The name of the organization that the *console server* belongs to.

  Locality/City The city where the organization is located.

  State/Province The state or province where the organization is located.

  Country The country where the organization is located. This is the two-letter ISO code, for example, DE for Germany, or US for the USA. (Note: Enter the country code in CAPITAL LETTERS.)

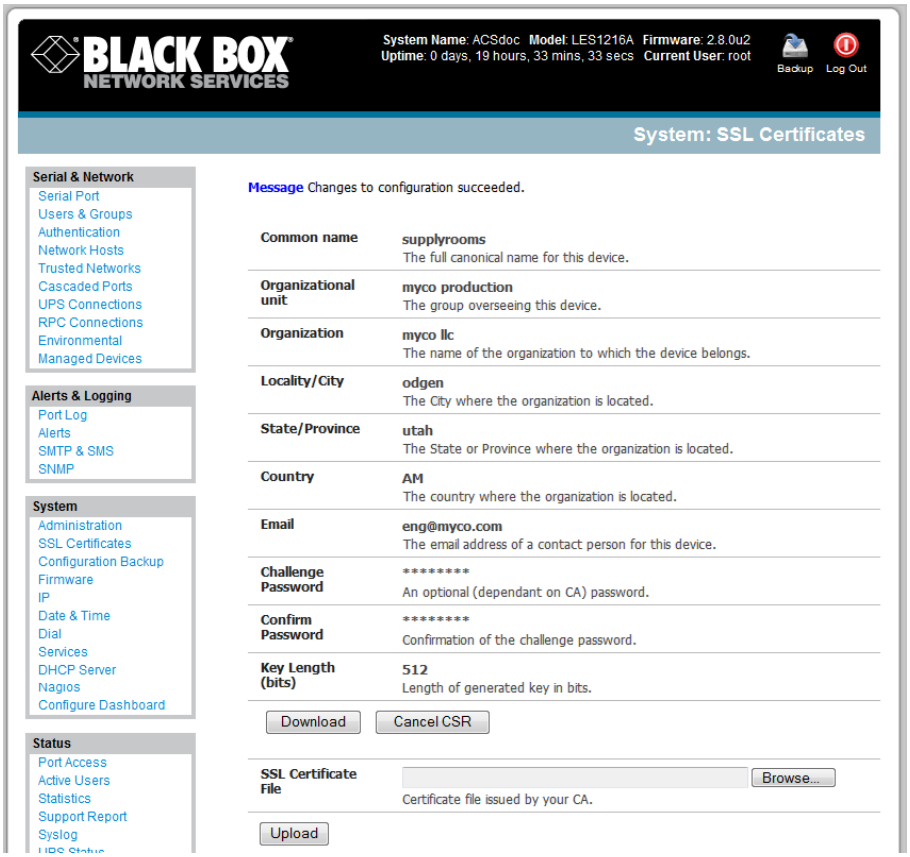
  Email The email address of a contact person that is responsible for the *console server* and its security.

  Challenge Password Some certification authorities require a challenge password to authorize later changes on the certificate (for example, revocation of the certificate). The password must be at least 4 characters long.

  Confirm Challenge Password Confirmation of the Challenge Password.

  Key length This is the length of the generated key in bits. 1024 Bits are supposed to be sufficient for most cases. Longer keys may result in slower response time of the *console server* when establishing connection.

- Once this is done, click on the button **Generate CSR** which will initiate the Certificate Signing Request generation. The CSR can be downloaded to your administration machine with the **Download** button.
- Send the saved CSR string to a Certification Authority (CA) for certification. You will get the new certificate from the CA after a more or less complicated traditional authentication process (depending on the CA).
- Upload the certificate to the *console server* using the **Upload** button as shown below.



After completing these steps, the *console server* has its own certificate that is used for identifying the *console server* to its users.

Note You can find information on issuing certificates and configuring HTTPS from the command line in Chapter 15.

Chapter 10 Nagios Integration

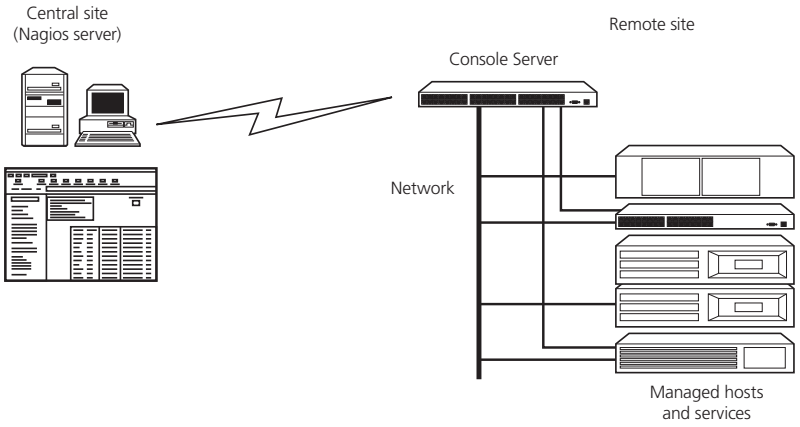
Introduction

Nagios is a powerful, highly extensible open source tool for monitoring network hosts and services. The core Nagios software package will typically be installed on a server or virtual server, the central Nagios server.

*Console server*s operate in conjunction with a central/upstream Nagios server to distribute and monitor attached network hosts and serial devices. They embed the NSCA (Nagios Service Checks Acceptor) and NRPE (Nagios Remote Plug-in Executor) add-ons—this allows them to communicate with the central Nagios server, so you won't need a dedicated slave Nagios server at remote sites.

The *console server* products all support basic distributed monitoring. Additionally, the Advanced Console Server (LES1208A, LES1216A, LES1248A) family supports extensive customizable distributed monitoring.

Even if distributed monitoring is not required, the *console server*s can be deployed locally alongside the Nagios monitoring host server, to provide additional diagnostics and points of access to managed devices.



SDT for Nagios extends the capabilities of the central Nagios server beyond monitoring, enabling it to be used for central management tasks. It incorporates the SDT Connector client, enabling point-and-click access and control of distributed networks of *console server*s and their attached network and serial hosts, from a central location.

Note If you have an existing Nagios deployment, you may want to use the *console server* gateways in a distributed monitoring server capacity only. If this case and you are already familiar with Nagios, skip ahead to section 10.3.

10.1 Nagios Overview

Nagios provides central monitoring of the hosts and services in your distributed network. Nagios is freely downloadable, open source software. This section offers a quick background of Nagios and its

capabilities. A complete overview, FAQ, and comprehensive documentation are available at: http://www.nagios.org

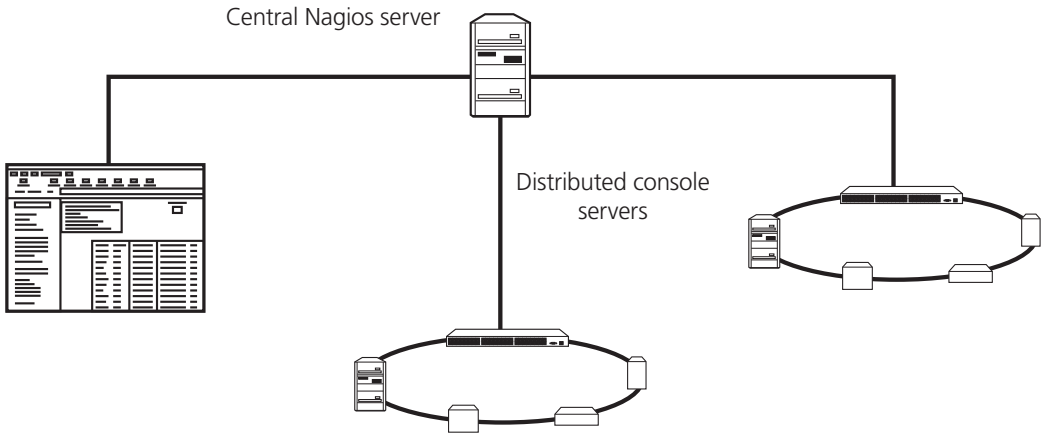
Nagios does take some time to install and configure, however once Nagios is up and running however, it provides an outstanding network monitoring system.

With Nagios you can:

- Display tables showing the status of each monitored server and network service in real time.
- Use a wide range of freely available plug-ins to make detailed checks of specific services—for example, don't just check that a database is accepting network connections, check that it can actually validate requests and return real data.
- Display warnings and send warning e-mails, pager, or SMS alerts when a service failure or degradation is detected.
- Assign contact groups who are responsible for specific services in specific time frames.

10.2 Central management and setting up SDT for Nagios

The Black Box Nagios solution has three parts: the Central Nagios server, Distributed Black Box *console server*s, and the SDT for Nagios software.



Central Nagios server

- A vanilla Nagios 2.x or 3.x installation (typically on a Linux server) generally running on a blade, PC, virtual machine, etc. at a central location.
- Runs a web server that displays the Nagios GUI.
- Imports configuration from distributed *console servers* using the SDT for Nagios Configuration Wizard.

Distributed *console server*s

- Black Box *console servers*.
- Serial and network hosts are attached to each *console server.*
- Each runs Nagios plug-ins, NRPE, and NSCA add-ons, but not a full Nagios server.

Clients

- Typically a client PC, laptop, etc., running Windows, Linux, or Mac OS X.
- Runs SDT Connector client software 1.5.0 or later.
- Possibly remote to the central Nagios server or distributed *console servers* (i.e. a road warrior).
- May receive alert emails from the central Nagios server or distributed *console servers.*
- Connects to the central Nagios server web UI to view status of monitored hosts and serial devices.
- Uses SDT Connector to connect through the *console servers* to manage monitored hosts and serial devices.

SDT Nagios setup involves the following steps:

i. Install Nagios and the NSCA and NRPE add-ons on the central Nagios server *(Section 10.2.1—Set up central Nagios server)*.

ii. Configure each Black Box distributed *console server* for Nagios monitoring, alerting, and SDT Nagios integration *(Section 10.2.2— Set up distributed* Black Box *servers).*

iii. Run the SDT for Nagios Configuration Wizard on the central Nagios server *(Section 10.2.3— Set up SDT Nagios on central Nagios server)* and perform any additional configuration tasks.

iv. Install SDT Connector on each client *(Section 10.2.4—Set up clients).*

10.2.1 Set up central Nagios server

SDT for Nagios requires a central Nagios server running Nagios 2.x or 3.x. Nagios 1.x is not supported. The Nagios server software is available for most major distributions of Linux using the standard package management tools. Your distribution will have documentation available on how to install Nagios. This is usually the quickest and simplest way to get up and running.

Note that you will need the core Nagios server package, and at least one of the NRPE or NSCA add-ons. NSCA is required to use the alerting features of the Black Box distributed hosts, installing both NRPE and NSCA is recommended.

You will also require a web server such as Apache to display the Nagios web UI (and this may be installed automatically depending on the Nagios packages).

Or, you may wish to download the Nagios source code directly from the Nagios website, and build and install the software from scratch. The Nagios website (http://www.nagios.org) has several Quick Start Guides that walk through this process.

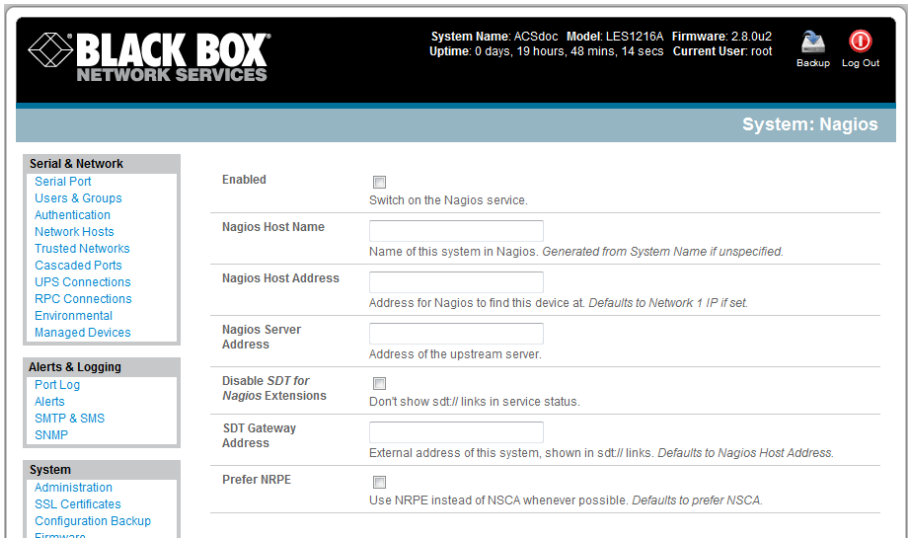
Once you are able to browse to your Nagios server and see its web UI and the local services it monitors by default, you are ready to continue.

10.2.2 Set up distributed *console server*s

This section provides a brief walkthrough on configuring a single *console server* to monitor the status of one attached network host (a Windows IIS server running HTTP and HTTPS services) and one serially

attached device (the console port of a network router), and to send alerts back to the Nagios server when an *Administrator* connects to the router or IIS server.

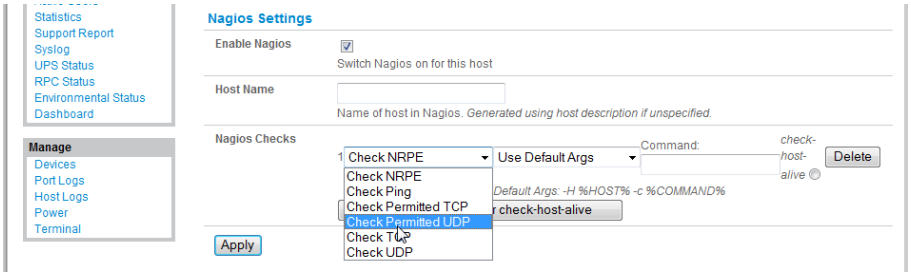
This walkthrough provides an example, but details of the configuration options are described in the next section. This walkthrough also assumes the network host and serial devices are already physically connected to the *console server*. The first step is to set up the Nagios features on the *console server*:



- Browse the Black Box *console server* and select **System: Nagios** on the *console server* Management Console. Check Nagios service **Enabled.**
- Enter the **Host Name** and the **Nagios Host Address** (for example, IP address) that the central Nagios server will use to contact the distributed Black Box *console server*.
- Enter the IP address that the distributed Black Box *console server* will use to contact the central Nagios server in **Nagios Server Address.**
- Enter the IP address that the clients running SDT Connector will use to connect through the distributed Black Box servers in **SDT Gateway address.**
- Check **Prefer NRPE**, **NRPE Enabled,** and **NRPE Command Arguments.**
- Check **NSCA Enabled**, choose an **NSCA Encryption Method** and enter and confirm an **NSCA Secret**. Remember these details because you will need them later on. For **NSCA Interval**, enter: *5*
- Click **Apply**.

Next, you must configure the attached Window network host and specify the services you will be checking with Nagios (HTTP and HTTPS):

- Select **Network Hosts** from the **Serial & Network** menu and click **Add Host**.
- Enter the **IP Address/DNS Name** of the network server, for example: *192.168.1.10* and enter a **Description**, for example: *Windows 2003 IIS Server.*
- Remove all **Permitted Services**. This server will be accessible using Terminal Services, so check **TCP**, **Port 3389** and log **level 1** and click **Add**. Remove and re-add the service to enable logging.



- Scroll down to **Nagios Settings** and check **Enable Nagios.**
- Click **New Check** and select **Check Ping**. Click **check-host-alive.**
- Click **New Check** and select **Check Permitted TCP**. Select **Port 3389**
- Click **New Check** and select **Check TCP**. Select **Port 80.**
- Click **New Check** and select **Check TCP**. Select **Port 443.**
- Click **Apply.**

Similarly, you now must configure the serial port to the router to be monitored by Nagios:

- Select **Serial Port** from the **Serial & Network** menu.
- Locate the serial port that has the router console port attached and click **Edit.**
- Make sure the serial port settings under *Common Settings* are correct and match the attached router's console port.
- Click ***Console server* Mode**, and select **Logging Level 1.**
- Check **Telnet** (SSH access is not required, as SDT Connector is used to secure the otherwise insecure Telnet connection).
- Scroll down to **Nagios Settings** and check **Enable Nagios.**
- Check **Port Log** and **Serial Status.**
- Click **Apply.**

Now you can set the *console server* to send alerts to the Nagios server:

- Select **Alerts** from the **Alerts & Logging** menu and click **Add Alert**.
- In **Description** enter: *Administrator connection.*
- Check **Nagios (NSCA).**
- In **Applicable Ports** check the serial port that has the router console port attached. In **Applicable Hosts** check the IP address/DNS name of the IIS server.
- Click **Connection Alert.**
- Click **Apply.**

Finally, you need to add a *User* for the client running SDT Connector:

- Select *Users & Groups* from the *Serial & Network* menu.
- Click **Add User.**

- In **Username**, enter: *sdtnagiosuser*, then enter and confirm a **Password.**
- In **Accessible Hosts** click the IP address/DNS name of the IIS server, and in **Accessible Ports** click the serial port that has the router console port attached.
- Click **Apply.**

10.3 Configuring Nagios distributed monitoring

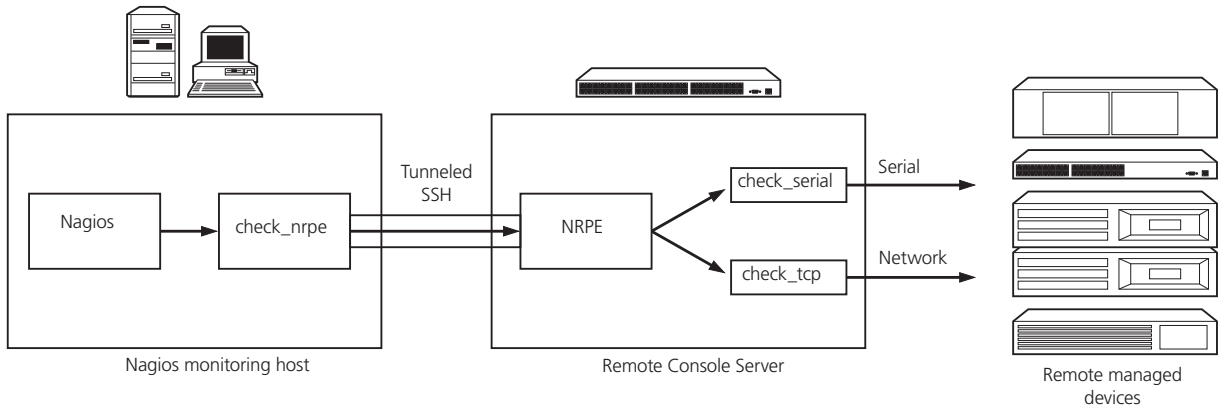
To activate the *console server* Nagios distributed monitoring:

- Nagios integration must be enabled and a path established to the central/upstream Nagios server.
- If the *console server* is to periodically report on Nagios monitored services, then the NSCA client embedded in the *console server* must be configured—the NSCA program enables scheduled check-ins with the remote Nagios server and is used to send passive check results across the network to the remote server.
- If the Nagios server is to actively request status updates from the *console server*, then the NRPE server embedded in the *console server* must be configured— the NRPE server is the Nagios daemon for executing plug-ins on remote hosts.
- Each of the Serial Ports and each of the Hosts connected to the *console server* that you want to monitor must have Nagios enabled and any specific Nagios checks configured.
- Configure the central/upstream Nagios monitoring host.

10.3.1 Enable Nagios on the *console server*

- Select **System: Nagios** on the *console server* Management Console and tick the Nagios service **Enabled.**
- Enter the **Nagios Host Name** that the *Console server* will be referred to in the Nagios central server—this will be generated from local System Name (entered in **System: Administration**) if unspecified.
- In **Nagios Host Address** enter the IP address or DNS name that the upstream Nagios server will use to reach the *console server*— if unspecified this will default to the first network port's IP (*Network (1)* as entered in **System: IP**).
- In **Nagios Server Address** enter the IP address or DNS name that the *console server* will use to reach the upstream Nagios monitoring server.
- Check the **Disable SDT Nagios Extensions** option if you want to disable the SDT Connector integration with your Nagios server at the head end— this would only be checked if you want to run a vanilla Nagios monitoring.
- If not, enter the IP address or DNS name that the SDT Nagios clients will use to reach the *console server* in **SDT Gateway Address.**
- When NRPE and NSCA are both enabled, NSCA is preferred method for communicating with the upstream Nagios server— check **Prefer NRPE** to use NRPE whenever possible (that is, for all communication except for alerts).

10.3.2 Enable NRPE monitoring

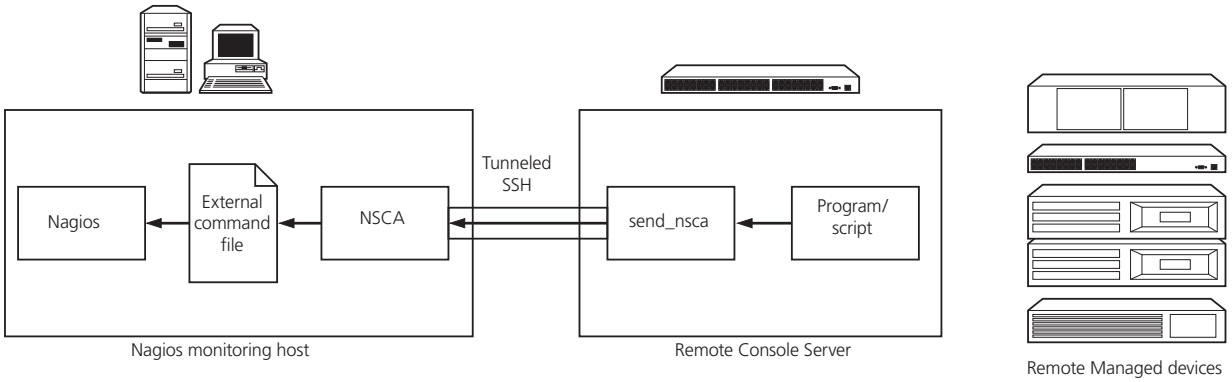


Enabling NRPE allows you to execute plug-ins (such as *check_tcp* and *check_ping*) on the remote *Console server* to monitor serial or network attached remote servers. This will offload CPU load from the upstream Nagios monitoring machine. This is especially valuable if you are monitoring hundreds or thousands of hosts. To enable NRPE:

- Select **System: Nagios** and check **NRPE Enabled**
- Enter the details for the user connection to the upstream Nagios monitoring server and again refer to the sample Nagios configuration example below for details about how to configure specific NRPE checks.

By default, the *console server* will accept a connection between the upstream Nagios monitoring server and the NRPE server with SSL encryption, without SSL, or tunneled through SSH. The security for the connection is configured at the Nagios server.

10.3.3 Enable NSCA monitoring



NSCA is the mechanism that allows you to send passive check results from the remote *console server* to the Nagios daemon running on the monitoring server. To enable NSCA:

- Select **System: Nagios** and check **NSCA Enabled.**

- Select the **Encryption** to be used from the drop down menu, then enter a **Secret** password and specify a check **Interval.**
- Refer to the sample Nagios configuration section below for some examples of configuring specific NSCA checks.

10.3.4 Configure Selected Serial Ports for Nagios Monitoring

The individual Serial Ports connected to the *console server* to be monitored must be configured for Nagios checks. Refer to *Chapter 4.4—Network Host Configuration* for details on enabling Nagios monitoring for Hosts that are network connected to the *console server*. To enable Nagios to monitor a device connected to the *console server* serial port:

- Select **Serial & Network: Serial Port** and click **Edit** on the serial Port # you want to monitor.
- Select **Enable Nagios,** specify the name of the device on the upstream server and determine the check you want to run on this port. **Serial Status** monitors the handshaking lines on the serial port and **Check Port** monitors the data logged for the serial port.

10.3.5 Configure Selected Network Hosts for Nagios Monitoring

The individual Network Hosts connected to the *console server* that you want to monitor must also be configured for Nagios checks:

- Select **Serial & Network: Network Port** and click **Edit** on the Network Host you want to monitor.
- Select **Enable Nagios,** specify the name of the device as it will appear on the upstream Nagios server.
- Click **New Check** to add a specific check which will be run on this host.
- Select **Check Permitted TCP/UDP** to monitor a service that you have previously added as a **Permitted Service.**
- Select **Check TCP/UDP** to specify a service port that you want to monitor, without allowing external (SDT Connector) access.
- Select **Check TCP** to monitor.
- The **Nagios Check** nominated as the ***check-host-alive*** check is the check used to determine whether the network host itself is up or down.
- Typically this will be *Check Ping*—although in some cases the host will be configured not to respond to pings.
- If no ***check-host-alive*** check is selected, the host will always be assumed to be up.
- You may deselect ***check-host-alive*** by clicking **Clear *check-host-alive.***
- If required, customize the selected **Nagios Checks** to use custom arguments.
- Click **Apply.**

10.3.6 Configure the upstream Nagios monitoring host

Refer to the Nagios documentation (http://www.nagios.org/docs/) for configuring the upstream server:

- The section entitled *Distributed Monitoring* steps through what you need to do to configure NSCA on the upstream server (under *Central Server Configuration*).
- *NRPE Documentation* was recently added that steps through configuring NRPE on the upstream server http://nagios.sourceforge.net/docs/nrpe/NRPE.pdf.

At this stage, Nagios at the upstream monitoring server is configured, and individual serial port and network host connections on the *console server* are configured for Nagios monitoring. If NSCA is enabled, each selected check will be executed once over the period of the check interval. If NRPE is enabled, then the upstream server will be able to request status updates under its own scheduling.

10.4 Advanced Distributed Monitoring Configuration

10.4.1 Sample Nagios configuration

An example configuration for Nagios is listed below. It shows how to set up a remote *Console server* to monitor a single host, with both network and serial connections. For each check it has two configurations, one each for NRPE and NSCA. In practice, these would be combined into a single check which used NSCA as a primary method, falling back to NRPE if a check was late— for details see the Nagios documentation (http://www.nagios.org/docs/) on *Service and Host Freshness Checks.*

```
; Host definitions
;
; Black Box console server
define host{
      use             generic-host
      host_name           Black Box
      alias           Console server
      address           192.168.254.147
      }

; Managed Host
define host{
      use             generic-host
      host_name           server
      alias           server
      address           192.168.254.227
      }

; NRPE daemon on gateway
define command {
          command_name          check_nrpe_daemon
          command_line  $USER1$/check_nrpe -H 192.168.254.147 -p 5666
          }

define service {
```

```
        service_description     NRPE Daemon
        host_name               Black Box
        use                     generic-service
        check_command                   check_nrpe_daemon
        }

; Serial Status
define command {
        command_name        check_serial_status
        command_line  $USER1$/check_nrpe -H 192.168.254.147 -p 5666 -c check_serial_$HOSTNAME$
        }

define service {
        service_description     Serial Status
        host_name               server
        use                     generic-service
        check_command                   check_serial_status
        }

define service {
        service_description     serial-signals-server
        host_name               server
        use                     generic-service
        check_command                   check_serial_status
        active_checks_enabled  0
        passive_checks_enabled          1
        }

define servicedependency{
        name                            Black Box_nrpe_daemon_dep
        host_name                       Black Box
        dependent_host_name             server
        dependent_service_description   Serial Status
        service_description             NRPE Daemon
        execution_failure_criteria      w,u,c
        }

; Port Log
define command{
        command_name    check_port_log
        command_line    $USER1$/check_nrpe -H 192.168.254.147 -p 5666 -c port_log_$HOSTNAME$
    }

define service {
        service_description     Port Log
        host_name               server
        use                     generic-service
        check_command                   check_port_log
```

```
        }

define service {
        service_description     port-log-server
        host_name               server
        use                     generic-service
        check_command                   check_port_log
        active_checks_enabled   0
        passive_checks_enabled          1
        }

define servicedependency{
        name                            Black Box_nrpe_daemon_dep
        host_name                       Black Box
        dependent_host_name             server
        dependent_service_description   Port Log
        service_description             NRPE Daemon
        execution_failure_criteria      w,u,c
        }

; Ping
define command{
        command_name    check_ping_via_Black Box
        command_line    $USER1$/check_nrpe -H 192.168.254.147 -p 5666 -c host_ping_$HOSTNAME$
    }

define service {
        service_description     Host Ping
        host_name               server
        use                     generic-service
        check_command                   check_ping_via_Black Box
        }

define service {
        service_description     host-ping-server
        host_name               server
        use                     generic-service
        check_command                   check_ping_via_Black Box
        active_checks_enabled   0
        passive_checks_enabled          1
        }

define servicedependency{
        name                            Black Box_nrpe_daemon_dep
        host_name                       Black Box
        dependent_host_name             server
        dependent_service_description   Host Ping
        service_description             NRPE Daemon
```

```
        execution_failure_criteria      w,u,c
        }

; SSH Port
define command{
    command_name    check_conn_via_Black Box
    command_line    $USER1$/check_nrpe -H 192.168.254.147 -p 5666 -c
host_$HOSTNAME$_$ARG1$_$ARG2$
    }

define service {
        service_description     SSH Port
        host_name               server
        use                     generic-service
        check_command                   check_conn_via_Black Box!tcp!22
        }

define service {
        service_description     host-port-tcp-22-server
                                ; host-port-<protocol>-<port>-<host>
        host_name               server
        use                     generic-service
        check_command                   check_conn_via_Black Box!tcp!22
        active_checks_enabled   0
        passive_checks_enabled          1
        }

define servicedependency{
        name                            Black Box_nrpe_daemon_dep
        host_name                       Black Box
        dependent_host_name             server
        dependent_service_description   SSH Port
        service_description             NRPE Daemon
        execution_failure_criteria      w,u,c
        }
```

10.4.2 Basic Nagios plug-ins

Plug-ins are compiled executables or scripts that can be scheduled to run on the *console server* to check the status of a connected host or service. This status is then communicated to the upstream Nagios server that uses the results to monitor the current status of the distributed network. Each *console server* is preconfigured with a selection of the checks that are part of the Nagios plug-ins package:

check_tcp and *check_udp* are used to check open ports on network hosts

check_ping is used to check network host availability

check_nrpe is used to execute arbitrary plug-ins in other devices

Each *console server* is preconfigured with two checks that are specific to Black Box:

check_serial_signals is used to monitor the handshaking lines on the serial ports

check_port_log is used to monitor the data logged for a serial port.

10.4.3 Additional plug-ins

Additional Nagios plug-ins (listed below) are available for Advanced Console Servers (LES1208A, LES1216A, LES1248A:

check_apt	*check_by_ssh*	*check_clamd*	*check_dig*
check_dns	*check_dummy*	*check_fping*	*check_ftp*
check_game	*check_hpjd*	*check_http*	*check_imap*
check_jabber	*check_ldap*	*check_load*	*check_mrtg*
check_mrtgtraf	*check_nagios*	*check_nntp*	*check_nntps*
check_nt	*check_ntp*	*check_nwstat*	*check_overcr*
check_ping	*check_pop*	*check_procs*	*check_real*
check_simap	*check_smtp*	*check_snmp*	*check_spop*
check_ssh	*check_ssmtp*	*check_swap*	*check_tcp*
check_time	*check_udp*	*check_ups*	*check_user*

You can download these plug-ins from the Nagios plug-ins package from *www.blackbox.com*.

You can also download and run *bash* scripts (primarily *check_log.sh).*

- To configure additional checks, save the downloaded plug-in program in the tftp *addins* directory on the USB flash and save the downloaded text plug-in file in */etc/config*
- To enable these new additional checks, select **Seria l& Network: Network Port**, then **Edit** the Network Host you want to monitor, and select **New Checks**. The additional check option is included in the updated **Nagios Checks** list, and you can again customize the arguments.

10.4.4 Number of supported devices

Ultimately the number of devices that by any particular *console server* can support depends upon the number of checks made, and how often they are performed. Access method will also play a part. The table below shows the performance of three of the *console servers*:

Time	No encryption	3DES	SSH tunnel
NSCA for single check	~ ½ second	~ ½ second	~ ½ second
NSCA for 100 sequential checks	100 seconds	100 seconds	100 seconds
NSCA for 10 sequential checks, batched upload	1 ½ seconds	2 seconds	1 second
NSCA for 100 sequential checks, batched upload	7 seconds	11 seconds	6 seconds

	No encryption	SSL	no encryption - tunneled over existing SSH session
NRPE time to service 1 check	1/10th second	1/3rd second	1/8th second
NRPE time to service 10 simultaneous checks	1 second	3 seconds	1 ¼ seconds
Maximum number of simultaneous checks before timeouts	30	20 (1,2 and 8) or 25 (16 and 48 port)	25 (8 port), 35 (16 and 48 port)

The results were from running tests 5 times in succession with no timeouts on any runs. There are a number of ways to increase the number of checks you can do.

Usually when using NRPE checks, an individual request will need to set up and tear down an SSL connection. This overhead can be avoided by setting up an SSH session to the *console server* and tunneling the NRPE port. This allows the NRPE daemon to run securely without SSL encryption, because SSH will provide the security.

When the *console server* submits NSCA results, it staggers them over a certain time period (for example, 20 checks over 10 minutes will result in two check results every minute). Staggering the results like this means that if the power fails or other incident causes multiple problems, the individual freshness checks will be staggered too.

NSCA checks are also batched. In the previous example, the two checks per minute are sent through in a single transaction.

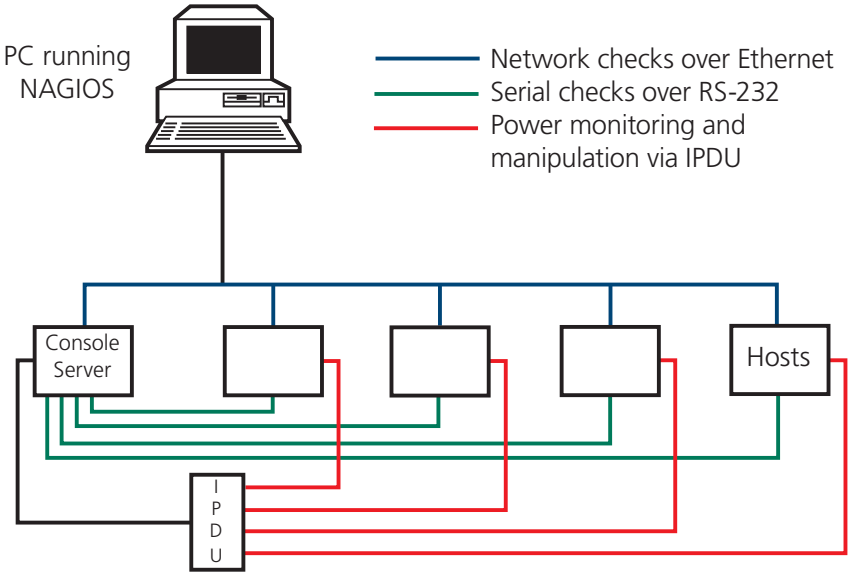
10.4.5 Distributed Monitoring Usage Scenarios

Below are a number of distributed monitoring Nagios scenarios:

I. Local office

In this scenario, the *console server* is set up to monitor each managed device's console. Configure it to make a number of checks, either actively at the Nagios server's request, or passively at preset intervals, and submit the results to the Nagios server in a batch.

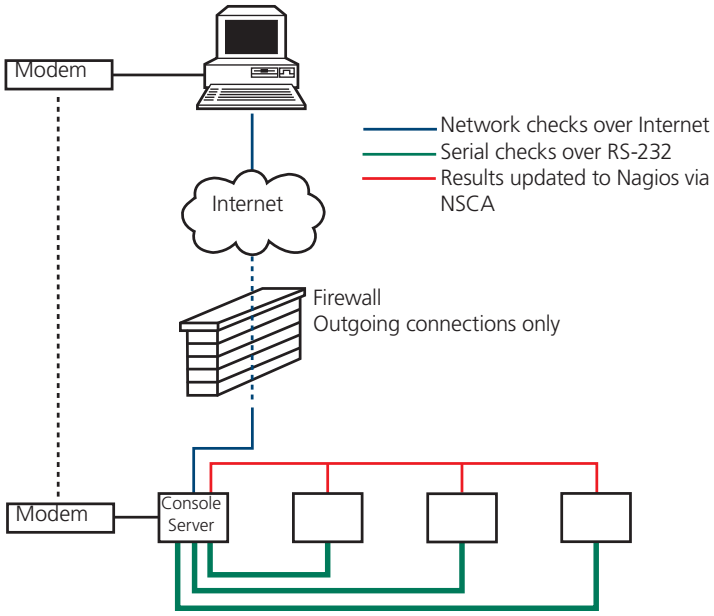
You can augment the *console server* at the local office site by one or more Intelligent Power Distribution Units (IPDUs) to remotely control the power supply to the managed devices.



II. Remote site

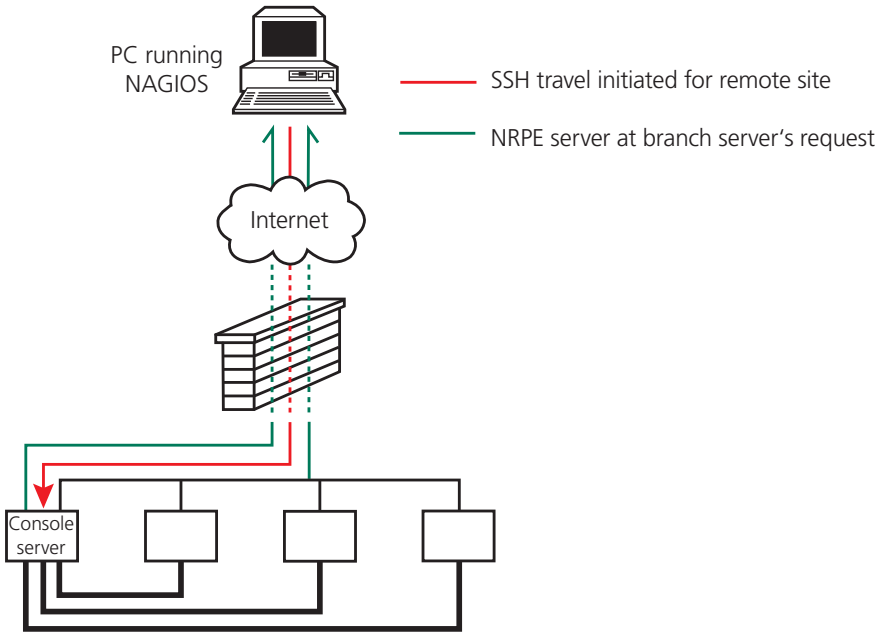
In this scenario, configure the *console server* NRPE server or NSCA client to actively check configured services and upload the checks to the Nagios server that's waiting passively. You can also configure it to service NRPE commands to perform checks on demand.

In this situation, the *console server* will perform checks based on both serial and network access.



Remote site with restrictive firewall

In this scenario, the role of the *console server* will vary. One aspect may be to upload check results through NSCA. Another may be to provide an SSH tunnel to allow the Nagios server to run NRPE commands.



Remote site with no network access

In this scenario the *console server* allows dial-in access for the Nagios server. Periodically, the Nagios server will establish a connection to the *console server* and execute any NRPE commands, before dropping the connection.

Chapter 11 System Management

Introduction

This chapter describes how the *Administrator* can perform a range of general *console server* system administration and configuration tasks such as:

- Applying *Soft* and *Hard* Resets to the gateway.
- Re-flashing the Firmware.
- Configuring the Date, Time and NTP.
- Setting up Backup of the configuration files.

System administration and configuration tasks that are covered elsewhere include:

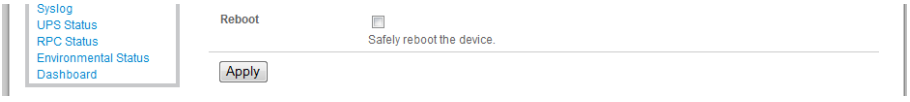
- Resetting the System Password and entering a new System Name and Description (*Chapter 3.2*).
- Setting the System IP Address (*Chapter 3.3).*
- Setting the permitted Services by which to access the gateway (*Chapter 3.4*).
- Setting up OoB Dial-in (*Chapter 5*).
- Configuring the Dashboard (*Chapter 12*).

11.1 System Administration and Reset

The *Administrator* can reboot or reset the gateway to default settings.

A *soft* reset is affected by:

- Selecting **Reboot** in the **System: Administration** menu and clicking **Apply.**



The *console server* reboots with all settings (*for example,* the assigned network IP address) preserved. This *soft* reset disconnects all users and ends any established SSH sessions.

A *soft* reset will also occur when you switch OFF power from the *console server*, and then switch the power back ON. If you cycle the power and the unit is writing to flash, you could corrupt or lose data, so rebooting the software is the safer option.

A *hard* erase (*hard reset*) is performed by:

- Pushing the *Erase* button on the rear panel **twice**. A ball-point pen or bent paper clip is a suitable tool for this procedure. Do not use a graphite pencil. Press the button gently **twice** (within a couple of seconds) while the unit is powered ON.

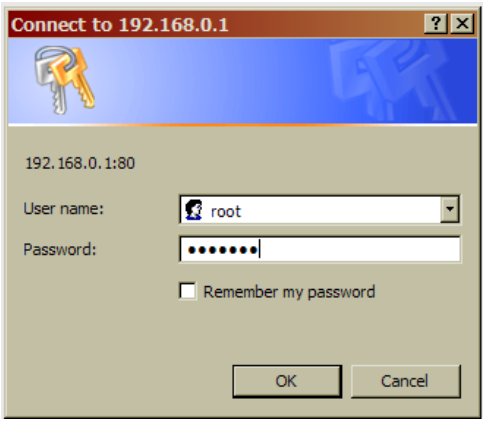
This will reset the *console server* back to its factory default settings and clear the *console server*'s stored configuration information.

The *hard* erase will clear all custom settings and return the unit back to factory default settings (*i.e.* the IP address will be reset to 192.168.0.1).

You will be prompted to log in and must enter the default administration username and administration password:

Username: **root**

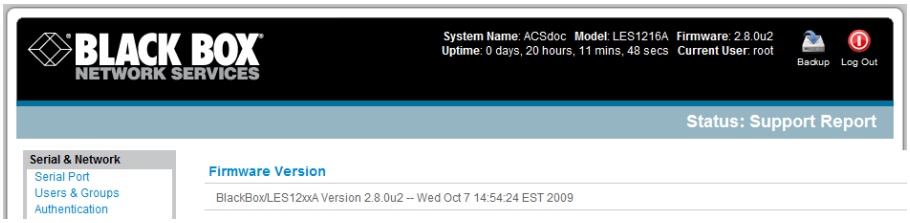
Password: **default**



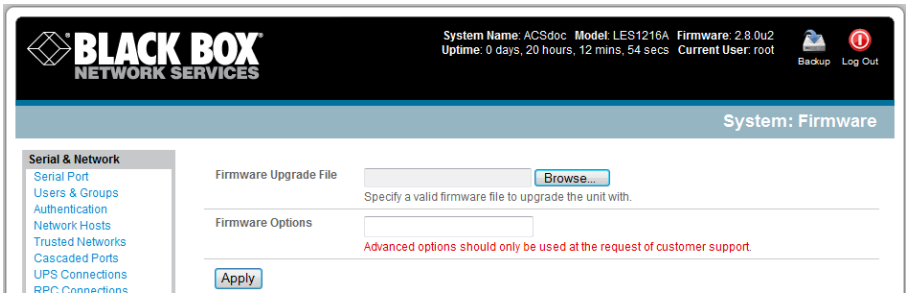
11.2 Upgrade Firmware

Before upgrading, make sure you are already running the most current firmware in your gateway. Your *console server* will not allow you to upgrade to the same or an earlier version.

- The **Firmware** version is displayed in each page's header.
- Or select **Status: Support Report** and note the **Firmware Version**.



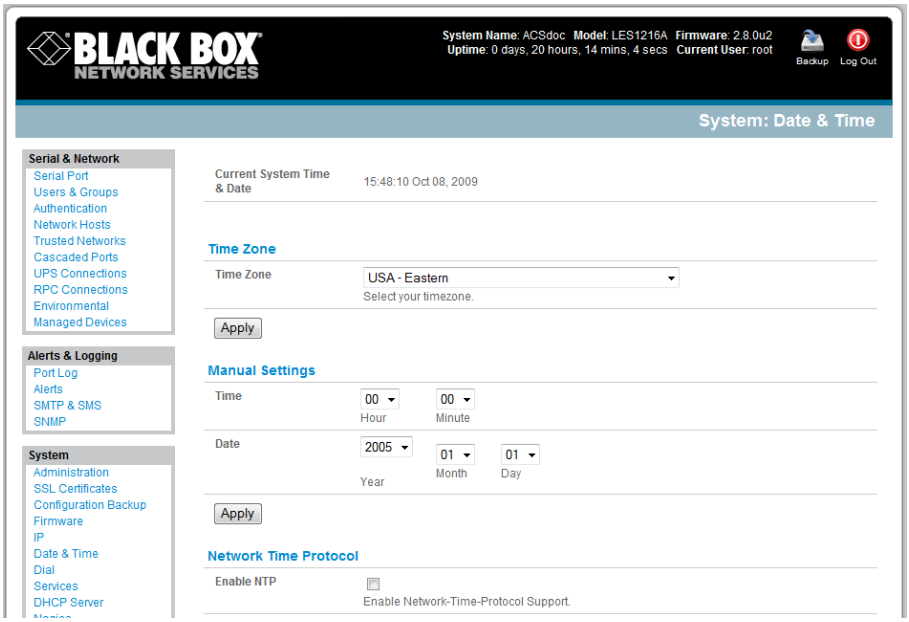
- To upgrade, you first must download the latest firmware image from the Black Box.web site.
- *S*ave this downloaded firmware image file to a system on the same subnet as the *console server.*
- Download and read the *release_notes.txt* for the latest information.
- To upload the firmware image file to your *console server*, select **System: Firmware**.
- Specify the address and name of the downloaded Firmware Upgrade File, or **Browse** the local subnet and locate the downloaded file.



- Click **Apply** and the *console server* appliance will perform a soft reboot and start upgrading the firmware. This process will take several minutes.
- After the firmware upgrade completes, click **here** to return to the Management Console. Your *console server* will have retained all its pre-upgrade configuration information.

11.3 Configure Date and Time

We recommend that you set the local Date and Time in the *console server* as soon as it is configured. Features like Syslog and NFS logging use the system time for time-stamping log entries, while certificate generation depends on a correct *Timestamp* to check the validity period of the certificate.



- Select the **System: Date & Time** menu option.
- Manually set the **Year**, **Month**, **Day**, **Hour** and **Minute** using the **Date** and **Time** selection boxes, then click **Apply.**

The gateway can synchronize its system time with a remote time server using the Network Time Protocol (NTP). Configuring the NTP time server ensures that the *console server* clock will be accurate soon after the Internet connection is established. Also if NTP is not used, the system clock will reset randomly every time the *console server* is powered up. To set the system time using NTP:

- Select the **Enable NTP** checkbox on the **Network Time Protocol** page.

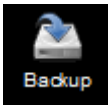
- Enter the IP address of the remote **NTP Server** and click **Apply.**

You must now also specify your local time zone so the system clock can show local time (and not UTP):

- Set your appropriate region/locality in the **Time Zone** selection box and click **Apply**.

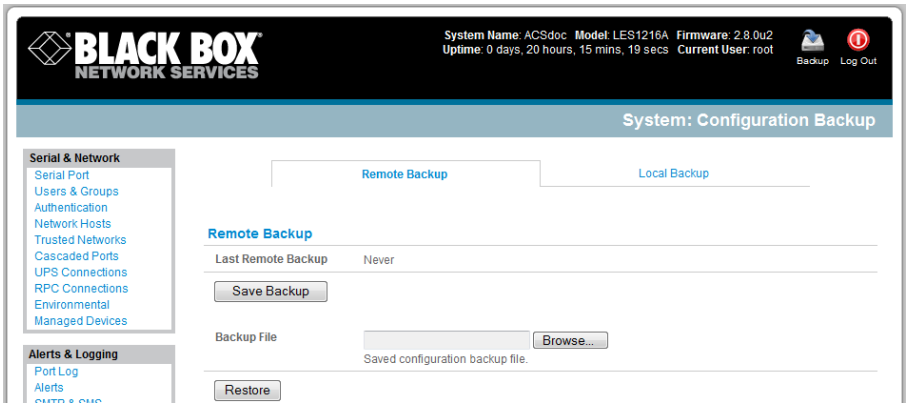
11.4 Configuration Backup

We recommend that you back up the *console server* configuration whenever you make significant changes (such as adding new Users or Managed Devices) or before performing a firmware upgrade.

- Select the **System: Configuration Backup** menu option or click the 

icon.

Note You can also back up the configuration files from the command line (refer to *Chapter 14*).



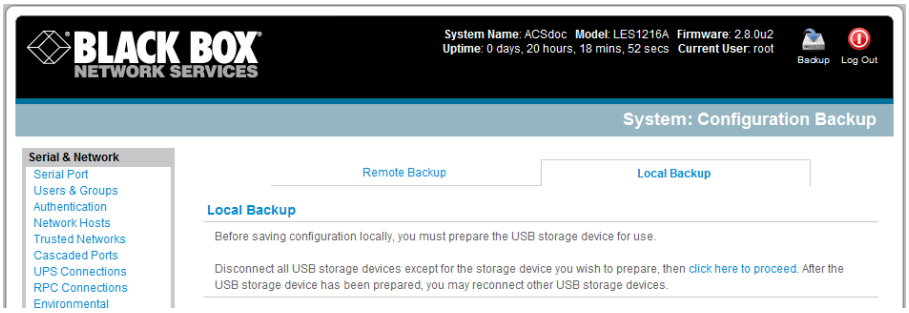
With all *console servers*, you can save the backup file remotely on your PC and you can restore configurations from remote locations:

- Click **Save Backup** in the Remote Configuration Backup menu.
- The config backup file (*System Name_date_config.opg*) will be downloaded to your PC and saved in the location you nominate.

To restore a remote backup:

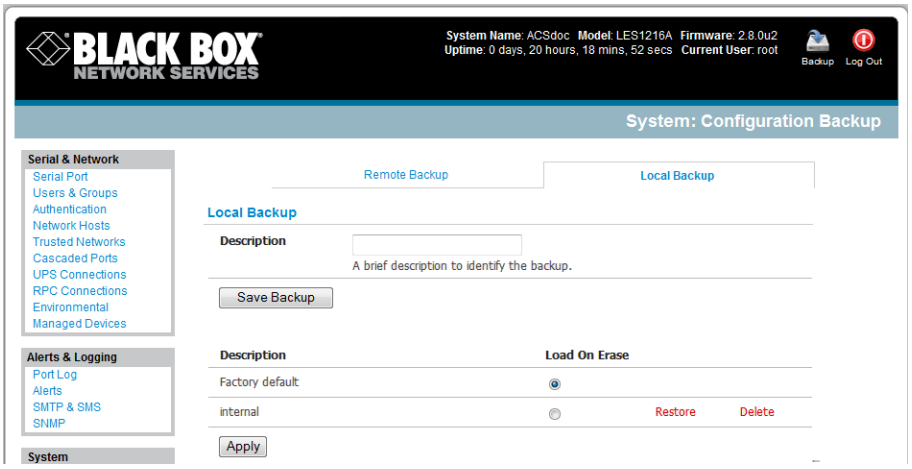
- Click **Browse** in the Remote Configuration Backup menu and select the **Backup File** you want to restore.
- Click **Restore** and click **OK.** This will overwrite all the current configuration settings in your *console server.*

With Advanced Console Servers (LES1208A, LES1216A, LES1248A), you can save the backup file locally on the *console server* USB storage. To do this you must have an external USB flash drive installed.


BLACK BOX
NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 20 hours, 18 mins, 52 secs Current User: root
Backup
Log Out
System: Configuration Backup
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Environmental
Remote Backup
Local Backup
Local Backup
Before saving configuration locally, you must prepare the USB storage device for use.
Disconnect all USB storage devices except for the storage device you wish to prepare, then click here to proceed. After the USB storage device has been prepared, you may reconnect other USB storage devices.

To backup and restore using USB:

- Make sure the USB flash is the only USB device attached to the *console server* and click **Prepare Storage** in the Local Configuration Backup menu.
- This will set a Volume Label on the USB storage device. This preparation step is only necessary the first time, and will not affect any other information you have saved onto the USB storage device. We recommend that you back up any critical data from the USB storage device before using it with your *console server.*
- If there are multiple USB devices installed, you will be warned to remove them.
- To backup to the USB, enter a brief **Description** of the backup in the Local Configuration Backups menu and select **Save Backup.**
- The Local Configuration Backup menu will display all the configuration backup files you have stored onto the USB flash.



- To restore a backup from the USB simply select **Restore** on the particular backup you wish to restore and click **Apply.**

After saving a local configuration backup, you may choose to use it as the alternate default configuration. When the *console server* is reset to factory defaults, it will then load your alternate default configuration instead of its factory settings:

- To set an alternate default configuration, check **Load On Erase** and click **Apply.**

Note: Before selecting *Load On Erase*, make sure that you have tested your alternate default configuration by clicking Restore.

If your alternate default configuration causes the *console server* to not boot, recover your unit to factory settings using the following steps:

- If the configuration is stored on an external USB storage device, unplug the storage device and reset to factory defaults as per section 11.1 of the user manual.
- If the configuration is stored on an internal USB storage device, reset it to factory defaults using a specially prepared USB storage device:
    - The USB storage device must be formatted with a Windows FAT32/VFAT file system on the first partition or the entire disk; most USB thumb drives are already formatted this way.
    - The file system must have the volume label: OPG_DEFAULT.
    - Insert this USB storage device into an external USB port on the *console server* and reset to factory defaults as described in Section 11.1.

- After recovering your *console server*, make sure the problem configuration is no longer selected for Load On Erase.

Introduction

This chapter describes the dashboard feature and the status reports that are available:

- Port Access and Active Users
- Statistics
- Support Reports
- Syslog
- Dashboard

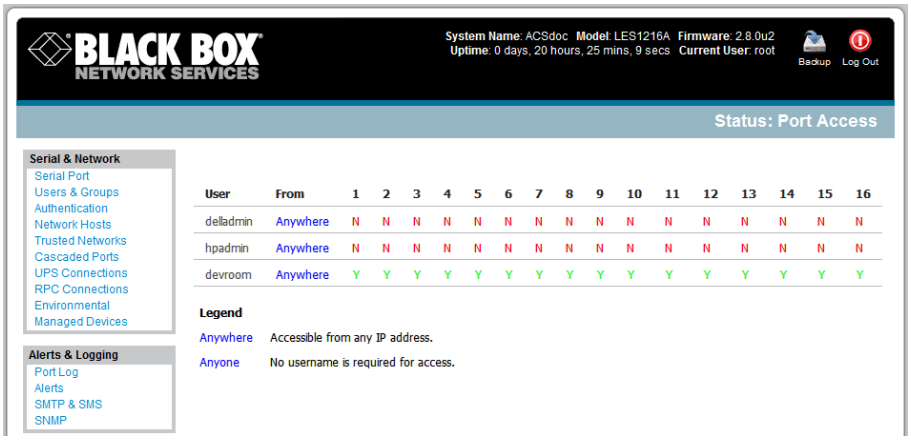
Other status reports that are covered elsewhere include:

- UPS Status (*Chapter 8.2*)
- RPC Status (*Chapter 8.1*)
- Environmental Status (*Chapter 8.3*)

12.1 Port Access and Active Users

The *Administrator* can see which *Users* have access privileges with which serial ports:

➢ Select the **Status: Port Access**


BLACK BOX
NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 20 hours, 25 mins, 9 secs Current User: root
Backup
Log Out
Status: Port Access
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Environmental
Managed Devices
Alerts & Logging
Port Log
Alerts
SMTP & SMS
SNMP
User From 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
delladmin Anywhere N N N N N N N N N N N N N N N N
hpadmin Anywhere N N N N N N N N N N N N N N N N
devroom Anywhere Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
Legend
Anywhere Accessible from any IP address.
Anyone No username is required for access.

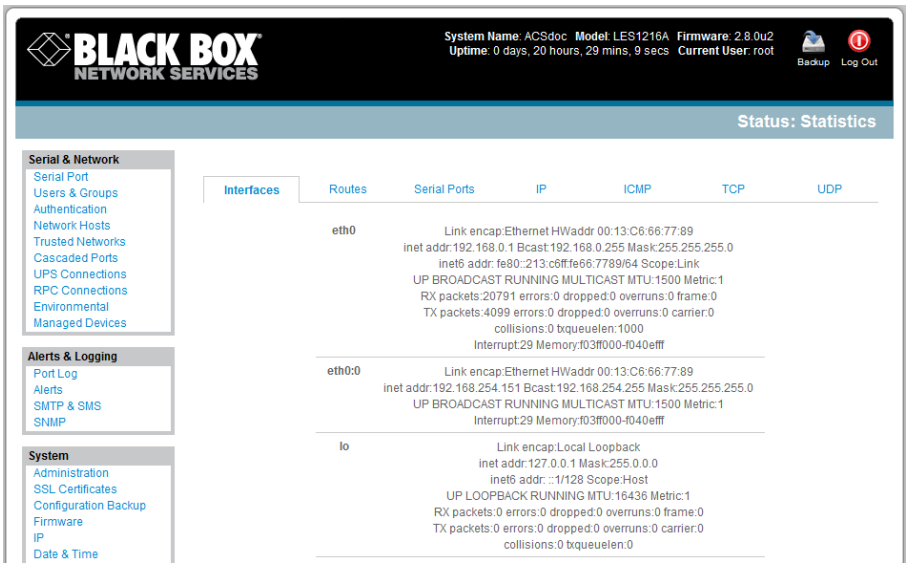
The *Administrator* can also see the current status as to *Users* who have active sessions on those ports:

➢ Select the **Status: Active Users**

12.2 Statistics

The Statistics report provides a snapshot of the status, current traffic, and other activities and operations of your *console server*:

- Select the **Status: Statistics**

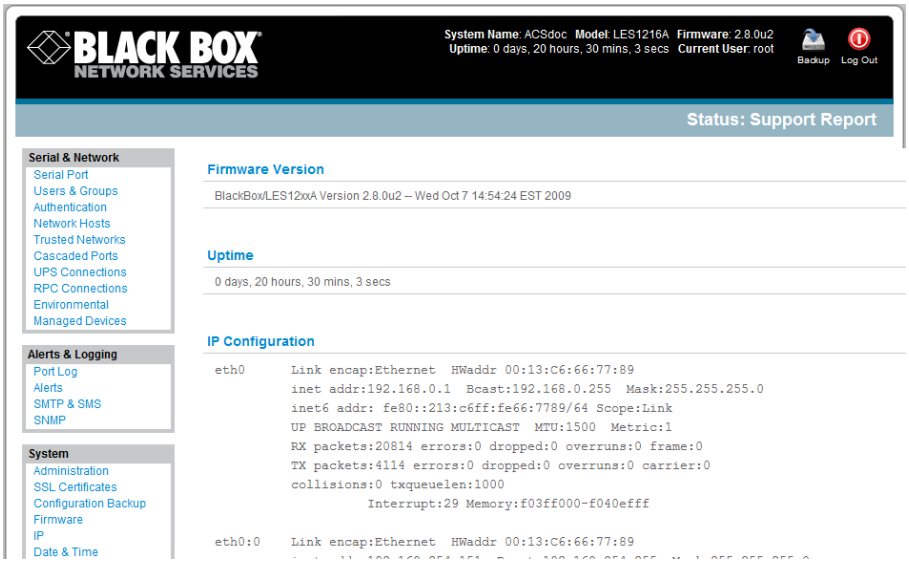


- You can find detailed statistics reports by selecting the various submenus.

12.3 Support Reports

The Support Report provides useful status information that will assist the Black Box Technical Support team to solve any problems you may experience with your *console server*.

If you do experience a problem and have to contact tech support, make sure you include the Support Report with your email support request. The Support Report is generated when the issue is occurring, and is attached in plain text format.



- Select **Status: Support Report** and you will be presented with a status snapshot.
- Save the file as a text file and attach it to your support email.

12.4 Syslog

The Linux System Logger in the *console server* maintains a record of all system messages and errors:

- Select **Status: Syslog**

You can redirect the syslog record to a remote Syslog Server:

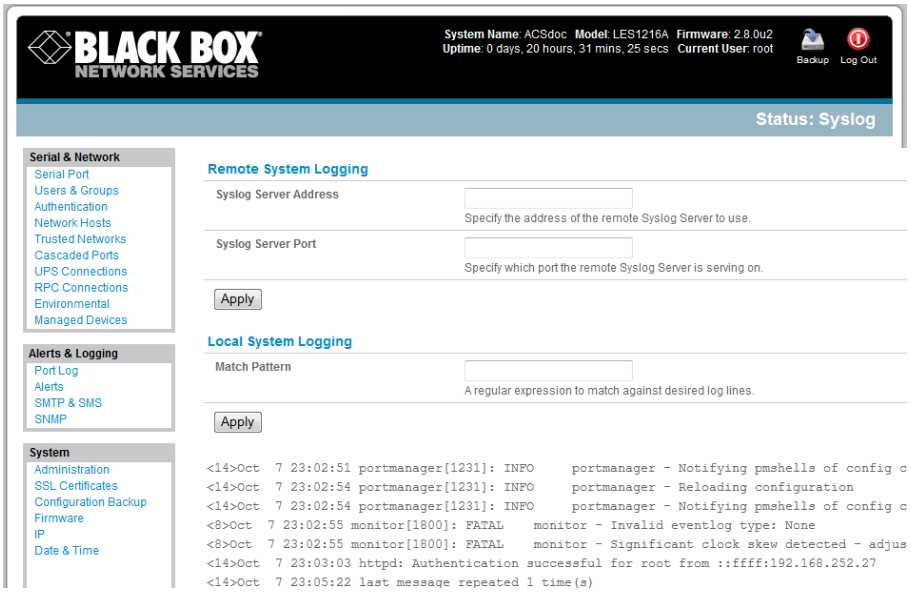
- Enter the remote **Syslog Server Address** and **Syslog Server Port** details and click **Apply.**

The console maintains a local Syslog. To view the local Syslog file:

- Select **Status: Syslog**

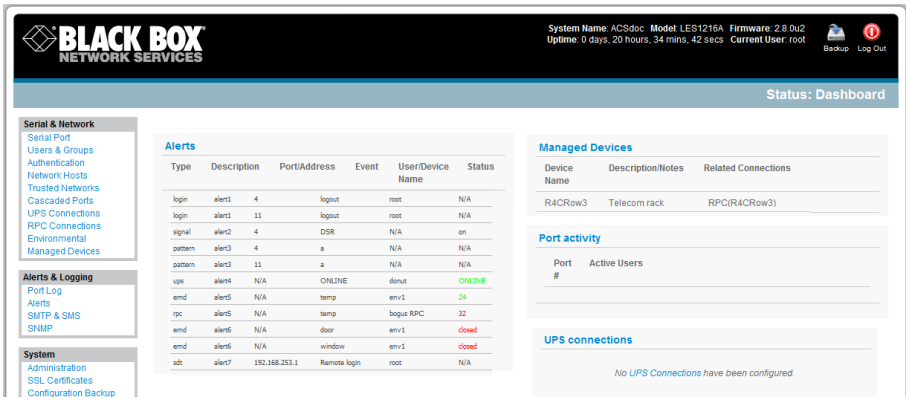
To make it easier to find information in the local Syslog file, use the provided pattern matching filter tool.

- Specify the **Match Pattern** that you want to search for (*for example,* the search for *mount* is shown below) and click **Apply.** The Syslog will then be represented with only those entries that actually include the specified pattern.


BLACK BOX
NETWORK SERVICES
System Name: ACSdoc Model: LES1216A Firmware: 2.8.0u2
Uptime: 0 days, 20 hours, 31 mins, 25 secs Current User: root
Backup
Log Out
Status: Syslog
Serial & Network
Serial Port
Users & Groups
Authentication
Network Hosts
Trusted Networks
Cascaded Ports
UPS Connections
RPC Connections
Environmental
Managed Devices
Alerts & Logging
Port Log
Alerts
SMTP & SMS
SNMP
System
Administration
SSL Certificates
Configuration Backup
Firmware
IP
Date & Time
Remote System Logging
Syslog Server Address
Specify the address of the remote Syslog Server to use.
Syslog Server Port
Specify which port the remote Syslog Server is serving on.
Apply
Local System Logging
Match Pattern
A regular expression to match against desired log lines.
Apply
<14>Oct 7 23:02:51 portmanager[1231]: INFO portmanager - Notifying pmshells of config c
<14>Oct 7 23:02:54 portmanager[1231]: INFO portmanager - Reloading configuration
<14>Oct 7 23:02:54 portmanager[1231]: INFO portmanager - Notifying pmshells of config c
<8>Oct 7 23:02:55 monitor[1800]: FATAL monitor - Invalid eventlog type: None
<8>Oct 7 23:02:55 monitor[1800]: FATAL monitor - Significant clock skew detected - adjus
<14>Oct 7 23:03:03 httpd: Authentication successful for root from ::ffff:192.168.252.27
<14>Oct 7 23:05:22 last message repeated 1 time(s)

12.5 Dashboard

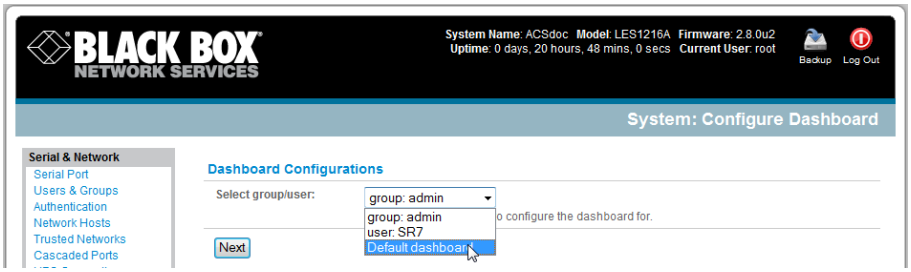
The Dashboard provides the *Administrator* with a summary of the status of the *console server* and its Managed Devices. You can configure custom dashboards for each user group.



12.5.1 Configuring the Dashboard

Only users who are members of the *admin* group (and the *root* user) can configure and access the dashboard. To configure a custom dashboard:

- Select **System: Configure Dashboard** and select the user (or group) you are configuring this custom dashboard layout for.
- Click **Next.**



Note: You can configure a custom dashboard for any *admin* user or for the *admin* group or you can reconfigure the default dashboard.

The *Status:Dashboard* screen is the first screen displayed when *admin* users (other than *root*) log into the console manager. If you log in as "*John*," and John is member of the *admin* group and there is a dashboard layout configured for John, then you will see the dashboard for John upon log-in and each time you click on the *Status:Dashboard* menu item.

If there is no dashboard layout configured for John, but there is an *admin* group dashboard configured, then you will see the admin group dashboard instead. If there is no user dashboard or admin group dashboard configured, then you will see the default dashboard.

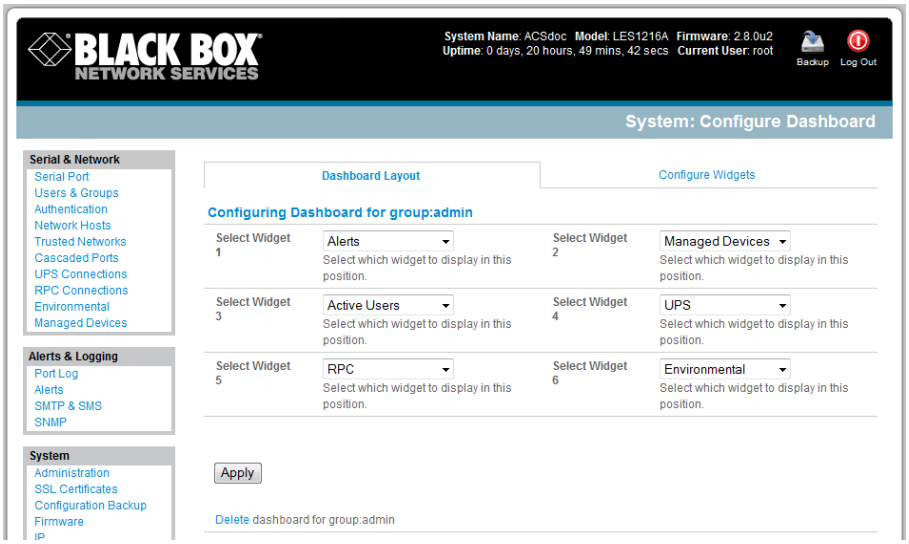
The *root* user does not have its own dashboard.

Use the above configuration options to enable admin users to setup their own custom dashboards.

The Dashboard displays six *widgets*. These widgets include each of the Status screens (alerts, devices, ports ups, rpc, and environmental status) and a custom script screen. The *admin* user can configure which of these widget is to be displayed where:

- Go to the **Dashboard layout** panel and select which widget is to be displayed in each of the six display locations (widget1 …6).

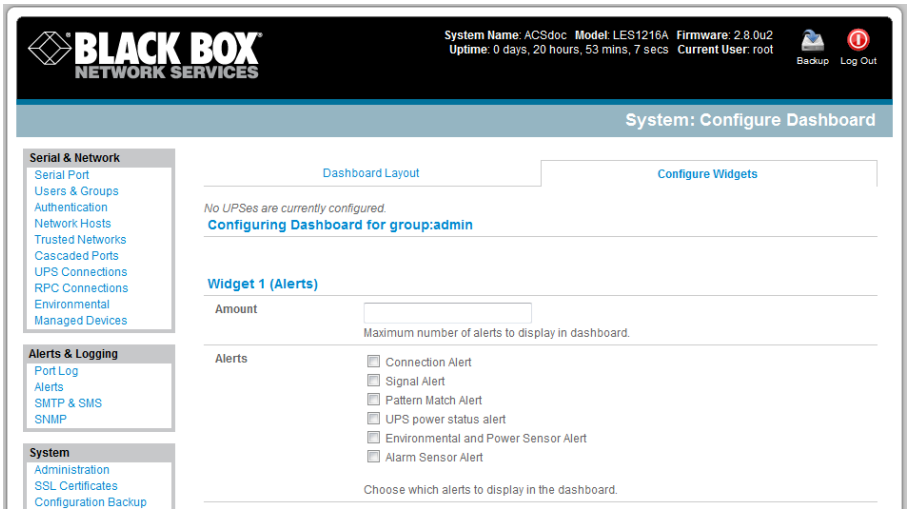
- Click **Apply.**



Note: The Alerts widget is a new screen that shows the current alerts status. When an alert gets triggered, a corresponding .XML file is created in */var/run/alerts/.* The dashboard scans all these files and displays a summary status in the alerts widget. When an alert is deleted, the corresponding .XML files that belong to that alert are also deleted.

To configure what is to be displayed by each widget:

- Go to the **Configure widgets** panel and configure each selected widget (for example, specify which UPS status is to be displayed on the *ups widget* or the maximum number of Managed Devices to be displayed in the *devices widget*.
- Click **Apply.**



Note: Dashboard configuration is stored in the /etc/config/config.xml file. Each configured dashboard will increase the config file. If this file gets too big, you can run out of memory space on the console manager.

12.5.2 Creating custom widgets for the Dashboard

T o run a custom script inside a dashboard widget:

Create a file called "*widget-<name>.sh*" in the folder */etc/config/scripts/* where <*name*> can be anything. You can have as many custom dashboard files as you want.

Inside this file you can put any code you want. When configuring the dashboard, choose "*widget-<name>.sh*" in the dropdown list. The dashboard will run the script and display the output of the script commands directly on the screen, inside the specific widget.

The best way to format the output would be to send HTML commands back to the browser by adding echo commands in the script:

echo '<table>'

You can of course run any command and its output will be displayed in the widget window directly. Below is an example script that writes the current date to a file, and then echos HTML code back to the browser. The HTML code gets an image from a specific URL and displays it in the widget.

```
#!/bin/sh

date >> /tmp/test
echo '<table>'
echo '<tr><td> This is my custom script running </td></tr>'
echo '<tr><td>'
echo '<img src="http://www.vinras.com/images/linux-online-inc.jpg">'
echo '</td></tr>'
echo '</table>'

exit 0
```

Chapter 13 Management

Introduction

The *console server* has a small number of **Manage** reports and tools that are available to both *Administrator*s and *Users*:

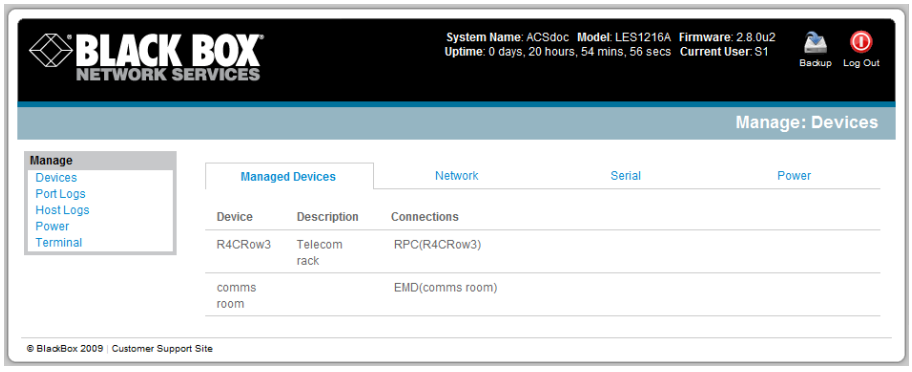
- Access and control authorized devices.
- View serial port logs and host logs for those devices.
- Use SDT Connector or the java terminal to access serially attached consoles.
- Control power devices (where authorized).

All other Management Console menu items are available to *Administrators* only.

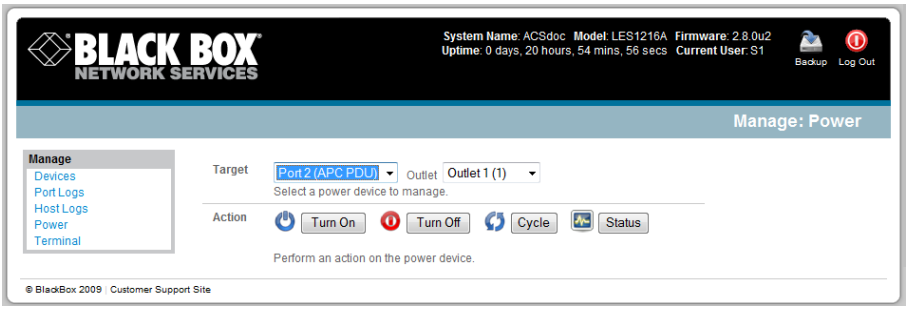
13.1 Device Management

To display the Managed Devices and their associated serial, network, and power connections:

- Select **Manage: Devices.** The *Administrator* will be presented with a list of all configured Managed Devices, whereas the *User* will only see the Managed Devices they (or their Group) has been given access privileges for.



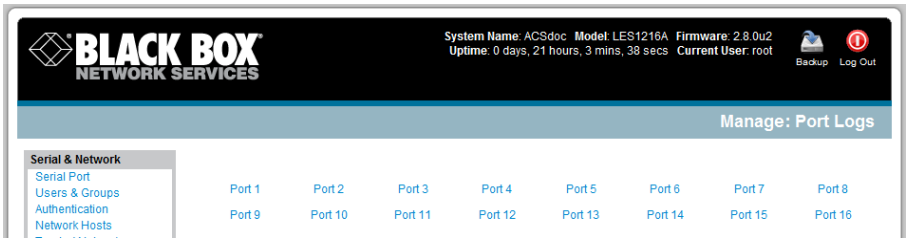
- Select **Serial Network** or **Power** for a view of the specific connections. The user can then take a range of actions using these serial, network or power connections by selecting the **Action** icon or the related Manage menu item. (For example, selecting the *Manager Power* icon [or **Manage: Power** from the menu] would enable the user to power Off/On/Cycle any power outlet on any PDU the user has been given access privileges to [refer to *Chapter 8* for details]).



13.2 Port and Host Logs

Administrators and *Users* can view logs of data transfers to connected devices.

- Select **Manage: Port Logs** and the serial Port # to be displayed.

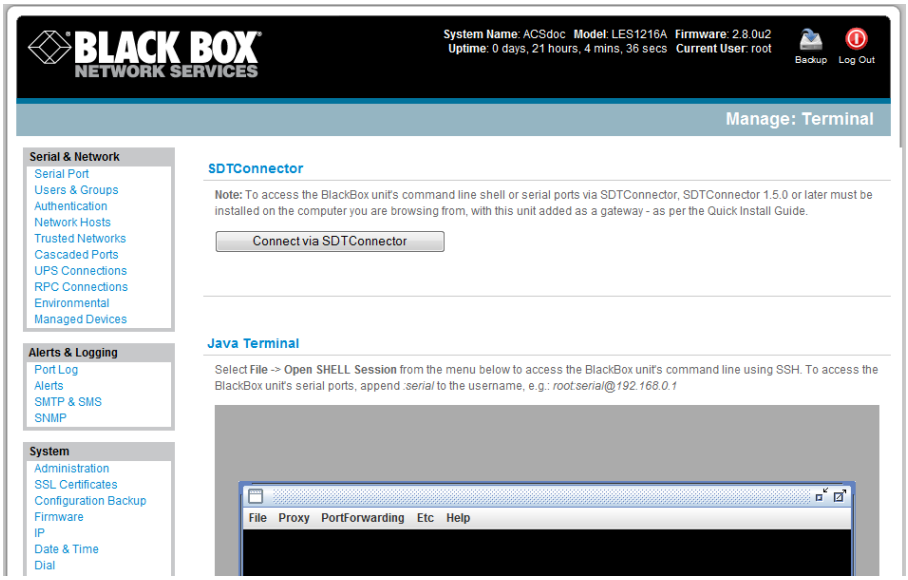


- To display Host logs, select **Manage: Host Logs** and the Host to be displayed.

13.3 Serial Port Terminal Connection

Administrator and *Users* can communicate directly with the *console server* command line and with devices attached to the *console server* serial ports using SDT Connector and their local tenet client, or use a java terminal in their browser.

- Select **Manage: Terminal.**



- Click **Connect to SDT Connector** to access the *console server's* command line shell or the serial ports via SDT Connector. This will to activate the SDT Connector client on the computer you are browsing from and load your local telnet client to connect to the command line or serial port using SSH.

Note You must install SDT Connector on the computer you are browsing from and add and the *console server* as a gateway as detailed in Chapter 6.

The alternate to using SDT Connector and your local telnet client is to run the open source *jcterm* java terminal applet into your browser to connect to the *console server* and attached serial port devices. *jcterm* does have some JRE compatibility issues that may prevent it from loading.

- Select **Manage: Terminal.** The *jcterm* java applet is downloaded from the *console server* to your browser and the virtual terminal will be displayed.
- Select **File** -> **Open SHELL Session** from the *jcterm* menu to access the command line using SSH.
- To access the *console server's* command line, enter its TCP address (e.g. *192.168.254.198*) as **hostname** and the Username, for example, *root@192.168.254.198.* Then enter the Password.
- To access the *console server*'s serial ports, append *:serial* to the username. With the gateway's TCP address (for example, *192.168.254.198*), the Username (for example, *root*), enter *root:serial@192.168.254.198.* Then enter Password and select the TCP Port address for the serial port to be accessed. By default *3001* is selected (that is, Port 1). To access Port 4 for example, change this to *3004* for the Username.

13.4 Power Management

Administrators and *Users* can access and manage the connected power devices.

- Select **Manage: Power**

Chapter 14 Command Line Configuration

Introduction

For those who prefer to configure their *console server* at the Linux command line level (rather than use a browser and the Management Console), this chapter describes how to use command line access and the ***config*** tool to manage the *console server* and configure the ports, etc.

This *config* documentation in this chapter walks through command line configuration to deliver the functions provided using the Management Console GUI.

For advanced and custom configurations and for details using other tools and commands, refer to the next chapter.

When displaying a command, the convention used in the rest of this chapter is to use single quotes ('') for user-defined values (for example, descriptions and names). Element values without single quotes must be typed exactly as shown.

After the initial section on accessing the *config* command, the menu items in this document follow the same structure as the menu items in the web GUI.

14.1 Accessing *config* from the command line

The *console server* runs a standard Linux kernel and embeds a suite of open source applications. If you do not want to use a browser and the Management Console tools, you can configure the *console server* and manage connected devices from the command line using standard Linux and Busybox commands and applications such as *ifconfig, gettyd, stty, powerman, nut etc*. Without care, these configurations may not withstand a *power-cycle-reset* or *reconfigure*.

Black Box provides a number of custom command line utilities and scripts to make it simple to configure the *console server* and make sure the changes are stored in the *console server*'s flash memory, etc.

In particular, the **config** utility allows you to manipulate the system configuration from the command line. With *config,* you can activate a new configuration by running the relevant configurator, which performs the action needed to make the configuration changes live.

To access *config* from the command line:

- Power on the *console server* and connect the “terminal” device:
  - If you are connecting using the serial line, plug a serial cable between the *console server* local DB9 console port and terminal device. Configure the serial connection of the terminal device you are using to 115200 bps, 8 data bits, no parity, and one stop bit.

- If you are connecting over the LAN, then you will need to interconnect the Ethernet ports and direct your terminal emulator program to the IP address of the *console server* (192.168.0.1 by default).

➢ Log on to the *console server* by pressing "return" a few times. The *console server* will request a username and password. Enter the username *root* and the password *default*. You should now see the command line prompt which is a hash (#).

This chapter is not intended to teach you Linux. We assume you already have a certain level of understanding before you execute Linux kernel level commands.

The *config* tool

Syntax

config [-ahv] [-d id] [-g id] [-p path] [-r configurator] [-s id=value] [-P id]

Description

The *config* tool is designed to perform multiple actions from one command if needed, so options can be chained together.

The *config* tool allows you to manipulate and query the system configuration from the command line. Using *config,* you can activate the new configuration by running the relevant *configurator* that performs the action needed to make the configuration changes live.

The custom user configuration is saved in the */etc/config/config.xml* file. This file is transparently accessed and edited when configuring the device using the Management Console browser GUI. Only the user *"root"* can configure from the shell.

By default, the *config* elements are separated by a '.' character. The root of the *config* tree is called <*config*>. To address a specific element place a '.' between each node/branch e.g. to access and display the description of *user1* type:

 # config -g config.users.user1.description

The root node of the *config* tree is <config>. To display the entire *config* tree, type:

 # config -g config

To display the help text for the *config* command, type:

 # config -h

The *config* application resides in the */bin* directory. The environmental variable called *PATH* contains a route to the */bin* directory. This allows a user to simply type *config* at the command prompt instead of the full path */bin/config*.

Options

-a –run-all	Run all registered configurators. This performs every configuration synchronization action pushing all changes to the live system
-h –help	Display a brief usage message

-v –verbose	Log extra debug information.
-d –del=id	Remove the given configuration element specified by a '.' separated identifier.
-g –get=id	Display the value of a configuration element.
-p –path=file	Specify an alternate configuration file to use. The default file is located at */etc/config/config.xml.*
-r –run=configurator	Run the specified registered configurator. Registered configurators are listed below.
-s --set=id=value	Change the value of configuration element specified by a '.' separated identifier.
-e --export=file	Save active configuration to file.
-i --import=file	Load configuration from file.
-t --test-import=file	Pretend to load configuration from file.
-S --separator=char	The pattern to separate fields with, default is '.'
-P --password=id	Prompt user for a value. Hash the value, then save it in id.

The registered configurators are:

alerts
auth
cascade
console
dhcp
dialin
eventlog
hosts
ipaccess
ipconfig
nagios
power
serialconfig
services
slave
systemsettings
time
ups
users

There are three ways to delete a config element value. The simplest way is use the *delete-node* script detailed later in Chapter 15. You can also assign the config element to "", or delete the entire config node using *-d*:

 # /bin/config -d 'element name'

All passwords are saved in plaintext *except* the user passwords and the system passwords, which are encrypted.

Note: The *config* command does not verify whether the nodes edited/added by the user are valid. This means that any node may be added to the tree. If a user runs the following command:

 # /bin/config -s config.fruit.apple=sweet

The configurator will not complain, but this command is useless. When the configurators are run (to turn the config.xml file into live config) they will simply ignore this <fruit> node. *Administrators* must make sure of the spelling when typing config commands. Incorrect spelling for a node will not be flagged.

Most configurations made to the XML file will be immediately active. To make sure that *all* configuration changes are active, especially when editing user passwords, run all the configurators:

 # /bin/config -a

For information on backing up and restoring the configuration file, refer to *Chapter 15, Advanced Configuration*.

14.2 Serial Port configuration

The first set of configurations you need to make to any serial port are the RS-232 common settings. For example, setup serial port 5 to use the following properties:

Baud Rate	*9600*
Parity	*None*
Data Bits	*8*
Stop Bits	*1*
label	*Myport*
log level	*0*
protocol	*RS232*
flow control	*None*

To do this, use the following commands:

 # config -s config.ports.port5.speed=9600
 # config -s config.ports.port5.parity=None
 # config -s config.ports.port5.charsize=8
 # config -s config.ports.port5.stop=1
 # config -s config.ports.port5.label=myport
 # config -s config.ports.port5.loglevel=0
 # config -s config.ports.port5.protocol=RS232
 # config -s config.ports.port5.flowcontrol=None

The following command will synchronize the live system with the new configuration:

 # config -r serialconfig

Note: Supported serial port baud-rates are ‘50’, ‘75’, ‘110’, ‘134’, ‘150’, ‘200’, ‘300’, ‘600’, ‘1200’, ‘1800’, ‘2400’, ‘4800’, ‘9600’, '19200', '38400', '57600', '115200', and '230400'.
Supported parity values are 'None', 'Odd', 'Even', 'Mark' and 'Space'.
Supported data-bits values are '8', '7', '6' and '5'.
Supported stop-bits values are '1', '1.5' and '2'.
Supported flow-control values are 'Hardware', 'Software' and 'None'.

Additionally, before any port can function properly, you need to set the port mode. Set any port to run in one of the five possible modes (refer *Chapter 4* for details): [*Console server* mode|Device mode|SDT mode|Terminal server mode|Serial bridge mode]. All these modes are mutually exclusive.

***Console server* mode**

The command to set the port in *portmanager* mode:

 # config -s config.ports.port5.mode=portmanager

To set the following optional config elements for this mode:

Data accumulation period	*100 ms*
Escape character	*% (default is ~)*
log level	*2 (default is 0)*
Shell power command menu	*Enabled*
RFC2217 access	*Enabled*
Limit pot to 1 connection	*Enabled*
SSH access	*Enabled*
TCP access	*Enabled*
telnet access	*Disabled*
Unauthorized telnet access	*Disabled*

 # config -s config.ports.port5.delay=100
 # config -s config.ports.port5.escapechar=%
 # config -s config.ports.port5.loglevel=2
 # config -s config.ports.port5.powermenu=on
 # config -s config.ports.port5.rfc2217=on
 # config -s config.ports.port5.singleconn=on
 # config -s config.ports.port5.ssh=on
 # config -s config.ports.port5.tcp=on
 # config -d config.ports.port5.telnet
 # config -d config.ports.port5.unauthtel

Device Mode

For a device mode port, set the port type to *ups, rpc,* or *enviro:*

 # config -s config.ports.port5.device.type=[ups | rpc | enviro]

For port 5 as a UPS port:

 # config -s config.ports.port5.mode=reserved

For port 5 as an RPC port:

 # config -s config.ports.port5.mode=powerman

For port 5 as an Environmental port:

 # config -s config.ports.port5.mode=reserved

SDT mode

To enable access over SSH to a host connected to serial port 5:

 # config -s config.ports.port5.mode=sdt
 # config -s config.ports.port5.sdt.ssh=on

To configure a username and password when accessing this port with Username = user1 and Password = secret:

 # config -s config.ports.port#.sdt.username=user1
 # config -s config.ports.port#.sdt.password=secret

Terminal server mode

Enable a TTY login for a local terminal attached to serial port 5:

 # config -s config.ports.port5.mode=terminal
 # config -s config.ports.port5.terminal=[vt220 | vt102 | vt100 | linux | ansi]

The default terminal is vt220.

Serial bridge mode

Create a network connection to a remote serial port via RFC-2217 on port 5:

 # config -s config.ports.port5.mode=bridge

Optional configurations for the network address of RFC-2217 server of 192.168.3.3 and TCP port used by the RFC-2217 service = 2500:

 # config -s config.ports.port5.bridge.address=192.168.3.3
 # config -s config.ports.port5.bridge.port=2500

To enable RFC-2217 access: *# config -s config.ports.port5.bridge.rfc2217=on*

To redirect the serial bridge over an SSH tunnel to the server: *# config -s config.ports.port5.bridge.ssh.enabled=on*

Syslog settings

Additionally, the global system log settings can be set for any specific port, in any mode:

 # config -s config.ports.port#.syslog.facility='facility'
'facility' can be:
- *Default*
- *local 0-7*
- *auth*
- *authpriv*
- *cron*
- *daemon*
- *ftp*
- *kern*
- *lpr*
- *mail*
- *news*
- *user*
- *uucp*

 # config -s config.ports.port#.syslog.priority='priority'
'priority' can be:
- *Default*
- *warning*
- *notice*
- *Info*
- *error*

emergency
debug
critical
alert

14.3 Adding and Removing Users

First, determine the total number of existing Users (if you have no existing Users you can assume this is 0):

 # config -g config.users.total

This command should display *config.users.total 1*. Note that if you see *config.users.total* this means you have 0 Users configured.

Your new User will be the existing total plus 1. If the previous command gave you 0, then you start with user number 1. If you already have 1 user your new user will be number 2, etc.

To add a user (with Username=John, Password=secret and Description =mySecondUser) issue the commands:

 # config -s config.users.total=2 (assuming we already have 1 user configured)
 # config -s config.users.user2.username=John
 # config -s config.users.user2.description=mySecondUser
 # config -P config.users.user2.password

NOTE: The -P parameter will prompt the user for a password, and encrypt it. You can encrypt the value of any config element using the -P parameter, but only encrypted user passwords and system passwords are supported. If any other element value were to be encrypted, the value will become inaccessible and will have to be reset.

To add this user to specific groups (admin/users):

 # config -s config.users.user2.groups.group1='groupname'
 # config -s config.users.user2.groups.group2='groupname2'
etc...

To give this user access to a specific port:

 # config -s config.users.user2.port1=on
 # config -s config.users.user2.port2=on
 # config -s config.users.user2.port5=on
etc...

To remove port access:

 # config -s config.users.user2.port1='' (the value is left blank)
or simply:
 # config -d config.users.user2.port1

The port number can be anything from 1 to 48, depending on the available ports on the specific *console server*.

For example, assume we have an RPC device connected to port 1 on the *console server* and the RPC is configured. To give this user access to RPC outlet number 3 on the RPC device, run the 2 commands below:

 # config -s config.ports.port1.power.outlet3.users.user2=John

 # config -s config.ports.port1.power.outlet3.users.total=2 (total number of users that have access to this outlet)

If more users are given access to this power outlet, then increment the *'config.ports.port1.power.outlet3.users.total'* element accordingly.

To give this user access to network host 5 (assuming the host is configured):

 # config -s config.sdt.hosts.host5.users.user1=John
 # config -s config.sdt.hosts.host5.users.total=1 (total number of users having access to host)

To give another user called "Peter" access to the same host:

 # config -s config.sdt.hosts.host5.users.user2=Peter
 # config -s config.sdt.hosts.host5.users.total=2 (total number of users having access to host)

To edit any of the user element values, use the same approach as when adding user elements, that is, use the "-s" parameter. If any of the config elements do not exist, they will automatically be created.

To delete the user called John, use the delete-node script:

 # ./delete-node config.users.user2

The following command will synchronize the live system with the new configuration:

 # config -r users

14.4 Adding and removing user Groups

The *console server* is configured with a few default user groups (even though only two of these groups are visible in the Management Console GUI). To find out how many groups are already present:

 # config -g config.groups.total

Assume this value is six. Make sure you number any new groups you create from seven and up.

To add a custom group to the configuration with Group name=Group7, Group description=MyGroup and Port access= 1,5 you'd issue the commands:

 # config -s config.groups.group7.name=Group7
 # config -s config.groups.group7.description=MyGroup
 # config -s config.groups.total=7
 # config -s config.groups.group7.port1=on
 # config -s config.groups.group7.port5=on

Assume we have an RPC device connected to port 1 on the console manager, and the RPC is configured. To give this group access to RPC outlet number 3 on the RPC device, run the two commands below:

 # config -s config.ports.port1.power.outlet3.groups.group1=Group7
 # config -s config.ports.port1.power.outlet3.groups.total=1 (total number of groups that have access to this outlet)

If more groups are given access to this power outlet, then increment the *'config.ports.port1.power.outlet3.groups.total*' element accordingly.

To give this group access to network host 5:

 # config -s config.sdt.hosts.host5.groups.group1=Group7
 # config -s config.sdt.hosts.host5.groups.total=1 (total number of groups having access to host)

To give another group called 'Group8' access to the same host:

 # config -s config.sdt.hosts.host5.groups.group2=Group8
 # config -s config.sdt.hosts.host5.groups.total=2 (total number of users having access to host)

To delete the group called Group7, use the following command:

 # rmuser Group7

Attention: The *rmuser* script is a generic script to remove any config element from config.xml correctly. However, any dependencies or references to this group will not be affected. Only the group details are deleted. The *Administrator* is responsible for going through *config.xml* and removing group dependencies and references manually, specifically if the group had access to a host or RPC device.

The following command will synchronize the live system with the new configuration:

 # config -a

14.5 Authentication

To change the type of authentication for the *console server*:

 # config -s config.auth.type='authtype'

'authtype' can be:

Local
LocalTACACS
TACACS
TACACSLocal
TACACSDownLocal
LocalRADIUS
RADIUS
RADIUSLocal
RADIUSDownLocal
LocalLDAP
LDAP
LDAPLocal
LDAPDownLocal

To configure TACACS authentication:

 # config -s config.auth.tacacs.auth_server='comma separated list' (list of remote authentiction and authorization servers.)
 # config -s config.auth.tacacs.acct_server='comma separated list' (list of remote accounting servers. If unset, Authentication and Authorization Server Address will be used.)
 # config -s config.auth.tacacs.password='password'

To configure RADIUS authentication:

 # config -s config.auth.radius.auth_server='comma separated list' (list of remote authentiction and authorization servers.)
 # config -s config.auth.radius.acct_server='comma separated list' (list of remote accounting servers. If unset, Authentication and Authorization Server Address will be used.)
 # config -s config.auth.radius.password='password'

To configure LDAP authentication:

 # config -s config.auth.ldap.server='comma separated list' (list of remote servers.)

 # config -s config.auth.ldap.basedn='name' (The distinguished name of the search base. For example: dc=my-company,dc=com)
 # config -s config.auth.ldap.binddn='name' (The distinguished name to bind to the server with. The default is to bind anonymously.)
 # config -s config.auth.radius.password='password'

The following command will synchronize the live system with the new configuration:

 # config -r auth

14.6 Network Hosts

To determine the total number of currently configured hosts:

 # config -g config.sdt.hosts.total

Assume this value is equal to 3. If you add another host, make sure you increment the total number of hosts from 3 to 4:

 # config -s config.sdt.hosts.total=4

If the output is *config.sdt.hosts.total* then assume 0 hosts are configured.

Add power device host

To add a UPS/RPC network host with the following details:

IP address/ DNS name	192.168.2.5
Host name	remoteUPS
Description	UPSroom3
Type	UPS
Allowed services	ssh port 22 and https port 443
Log level for services	0

Issue the commands below:

 # config -s config.sdt.hosts.host4.address=192.168.2.5
 # config -s config.sdt.hosts.host4.name=remoteUPS
 # config -s config.sdt.hosts.host4.description=UPSroom3
 # config -s config.sdt.hosts.host4.device.type=ups
 # config -s config.sdt.hosts.host4.tcpports.tcpport1=22
 # config -s config.sdt.hosts.host4.tcpports.tcpport1.loglevel=0
 # config -s config.sdt.hosts.host4.udpports.udpport2=443
 # config -s config.sdt.hosts.host4.udpports.udpport2.loglevel=0

The *loglevel* can have a value of 0 or 1.

The default services that you should configure are: *22/tcp (ssh), 23/tcp (telnet), 80/tcp (http), 443/tcp (https), 1494/tcp (ica), 3389/tcp (rdp), 5900/tcp (vnc)*

Add other network host

To add any other type of network host with the following details:

IP address/ DNS name	192.168.3.10
Host name	OfficePC
Description	MyPC
Allowed sevices	ssh port 22,https port 443
log level for services	1

Issue the commands below. If the Host is not a PDU or UPS power device or a server with IPMI power control, then leave the device type blank:

 # config -s config.sdt.hosts.host4.address=192.168.3.10
 # config -s config.sdt.hosts.host4.description=MyPC
 # config -s config.sdt.hosts.host4.name=OfficePC
 # config -s config.sdt.hosts.host4.device.type='' (leave this value blank)
 # config -s config.sdt.hosts.host4.tcpports.tcpport1=22
 # config -s config.sdt.hosts.host4.tcpports.tcpport1.loglevel=1
 # config -s config.sdt.hosts.host4.udpports.tcppport2=443
 # config -s config.sdt.hosts.host4.udpports.tcpport2.loglevel=1

If you want to add the new host as a managed device, make sure you use the current total number of managed devices + 1, for the new device number.

To get the current number of managed devices:

 # config -g config.devices.total

Assuming we already have one managed device, our new device will be device 2. Issue the following commands:

 # config -s config. devices.device2.connections.connection1.name=192.168.3.10
 # config -s config. devices.device2.connections.connection1.type=Host
 # config -s config. devices.device2.name=OfficePC
 # config -s config. devices.device2.description=MyPC
 # config -s config.devices.total=2

The following command will synchronize the live system with the new configuration:

 # config -hosts

14.7 Trusted Networks

You can further restrict remote access to serial ports based on the source IP address. To configure this via the command line, you need to do the following:

Determine the total number of existing trusted network rules. If you have no existing rules, you can assume this is 0.

 # config -g config.portaccess.total

This command should display *config.portaccess.total 1*

Note that if you *see config.portaccess.total* this means you have 0 rules configured.

Your new rule will be the existing total plus 1. So if the previous command gave you 0, then you start with rule number 1. If you already have 1 rule your new rule will be number 2, etc.

If you want to restrict access to serial port 5 to computers from a single class C network (*192.168.5.0* for example), you need to issue the following commands (assuming you have a previous rule in place).

Add a trusted network:

 # config -s config.portaccess.rule2.address=192.168.5.0
 # config -s "config.portaccess.rule2.description=foo bar"
 # config -s config.portaccess.rule2.netmask=255.255.255.0
 # config -s config.portaccess.rule2.port5=on
 # config -s config.portaccess.total=2

The following command will synchronize the live system with the new configuration:

 # config -r serialconfig

14.8 Cascaded Ports

To add a new slave device with the following settings:

IP address/DNS name	192.168.0.153
Description	Console in office 42
Label	les1116-5
Number of ports	16

The following commands must be issued:

 # config -s config.cascade.slaves.slave1.address=192.168.0.153
 # config -s "config.cascade.slaves.slave1.description=CM in office 42"
 # config -s config.cascade.slaves.slave1.label=les1116-5
 # config -s config.cascade.slaves.slave1.ports=16

The total number of slaves must also be incremented. If this is the first slave you're adding, type:

 # config -s config.cascade.slaves.total=1

Increment this value when adding more slaves.

NOTE: If a slave is added using the CLI, then the master SSH public key will need to be manually copied to every slave device before cascaded ports will work (refer *Chapter 4*).

The following command will synchronize the live system with the new configuration:

 # config -r cascade

14.9 UPS Connections

Managed UPSes

Before adding a managed UPS, make sure that at least 1 port has been configured to run in 'device mode', and that the device is set to 'ups'.

To add a managed UPS with the following values:

Connected via	Port 1
UPS name	My UPS
Description	UPS in room 5
Username to connect to UPS	User2
Password to connect to UPS	secret
shutdown order	2 (0 shuts down first)
Driver	genericups
Driver option - option	option
Driver option - argument	argument
Logging	Enabled
Log interval	2 minutes
Run script when power is critical	Enabled

 # config -s config.ups.monitors.monitor1.port=/dev/port01
If the port number is higher than 9, eg port 13, enter:
 # config -s config.ups.monitors.monitor1.port=/dev/port13

 # config -s "config.ups.monitors.monitor1.name=My UPS"

 # config -s "config.ups.monitors.monitor1.description=UPS in room 5"
 # config -s config.ups.monitors.monitor1.username=User2
 # config -s config.ups.monitors.monitor1.password=secret
 # config -s config.ups.monitors.monitor1.sdorder=2
 # config -s config.ups.monitors.monitor1.driver=genericups
 # config -s config.ups.monitors.monitor1.options.option1.opt=option
 # config -s config.ups.monitors.monitor1.options.option1.arg=argument
 # config -s config.ups.monitors.monitor1.options.total=1
 # config -s config.ups.monitors.monitor1.log.enabled=on
 # config -s config.ups.monitors.monitor1.log.interval=2
 # config -s config.ups.monitors.monitor1.script.enabled=on

Make sure to increment the total monitors:

 # config -s config.ups.monitors.total=1

The five commands below will add the UPS to Managed devices. Assuming there are already two managed devices configured:

 # config -s "config.devices.device3.connections.connection1.name=My UPS"
 # config -s "config.devices.device3.connections.connection1.type=UPS Unit"
 # config -s "config.devices.device3.name=My UPS"
 # config -s "config.devices.device3.description=UPS in toom 5"
 # config -s config.devices.total=3

To delete this managed UPS:

 # config -d config.ups.monitors.monitor1

Decrement *monitors.total* when deleting a managed UPS.

Remote UPSes

To add a remote UPS with the following details (assuming this is our first remote UPS):

UPS name	oldUPS
Description	UPS in room 2
Address	192.168.50.50
Log status	Disabled
Log rate	240 seconds
Run shutdown script	Enabled

 # config -s config.ups.remotes.remote1.name=oldUPS
 # config -s "config.ups.remotes.remote1.description=UPS in room 2"
 # config -s config.ups.remotes.remote1.address=192.168.50.50
 # config -d config.ups.remotes.remote1.log.enabled
 # config -s config.ups.remotes.remote1.log.interval=240
 # config -s config.ups.remotes.remote1.script.enabled=on
 # config -s config.ups.remotes.total=1

The following command will synchronize the live system with the new configuration:

 # config -a

14.10 RPC Connections

You can add an RPC connection from the command line. We do not recommend that you do this because of dependency issues.

However FYI before adding an RPC the Management Console GUI code makes sure that at least one port has been configured to run in 'device mode', and that the device is set to 'rpc'.

To add an RPC with the following values:

RPC type	APC 7900
Connected via	Port 2
UPS name	MyRPC
Description	RPC in room 5
Login name for device	rpclogin
Login password for device	secret
SNMP community	v1 or v2c
Logging	Enabled
Log interval	600 second
Number of power outlets	4 (depends on the type/model of the RPC)

 # config -s config.ports.port2.power.type=APC 7900
 # config -s config.ports.port2.power.name=MyRPC
 # config -s "config.ports.port2.power.description=RPC in room 5"
 # config -s config.ports.port2.power.username=rpclogin
 # config -s config.ports.port2.power.password=secret
 # config -s config.ports.port2.power.snmp.community=v1
 # config -s config.ports.port2.power.log.enabled=on
 # config -s config.ports.port2.power.log.interval=600
 # config -s config.ports.port2.power.outlets=4

The following five commands are used by the Management Console to add the RPC to “Managed Devices”:

 # config -s config.devices.device3.connections.connection1.name=myRPC
 # config -s "config.devices.device3.connections.connection1.type=RPC Unit"
 # config -s config.devices.device3.name=myRPC
 # config -s "config.devices.device3.description=RPC in room 5"
 # config -s config.devices.total=3

The following command will synchronize the live system with the new configuration:

 # config -a

14.11 Environmental

To configure an environmental monitor with the following details:

Monitor name	Envi4
Monitor Description	Monitor in room 5
Temperature offset	2
Humidity offset	5
Enable alarm 1 ?	yes
Alarm 1 label	door alarm
Enable alarm 2 ?	yes
Alarm 2 label	window alarm
Logging enabled ?	yes
Log interval	120 seconds

 # config -s config.ports.port3.enviro.name=Envi4
 # config -s "config.ports.port3.enviro.description=Monitor in room 5"

 # config -s config.ports.port3.enviro.offsets.temp=2
 # config -s config.ports.port3.enviro.offsets.humid=5
 # config -s config.ports.port3.enviro.alarms.alarm1.alarmstate=on
 # config -s config.ports.port3.enviro.alarms.alarm1.label=door alarm
 # config -s config.ports.port3.enviro.alarms.alarm2.alarmstate=on
 # config -s config.ports.port3.enviro.alarms.alarm2.label=window alarm
 # config -s config.ports.port3.enviro.alarms.total=2
 # config -s config.ports.port3.enviro.log.enabled=on
 # config -s config.ports.port3.enviro.log.interval=120

Assign alarms.total=2 even if they are off.

The following 5 commands will add the environmental monitor to “Managed devices”:

To get the total number of managed devices:

 # config -g config.devices.total

Make sure you use the total + 1 for the new device below:

 # config -s config. devices.device5.connections.connection1.name=Envi4
 # config -s "config. devices.device5.connections.connection1.type=EMD Unit"
 # config -s config. devices.device5.name=Envi4
 # config -s "config. devices.device5.description=Monitor in room 5"
 # config -s config.devices.total=5

The following command will synchronize the live system with the new configuration:

 # config -a

14.12 Managed Devices

To add a managed device: (also see UPS, RPC connections and Environmental)

 # config -s "config.devices.device8.name=my device"
 # config -s "config.devices.device8.description=The eighth device"
 # config -s "config.devices.device8.connections.connection1.name=my device"
 # config -s config.devices.device8.connections.connection1.type=[serial | Host | UPS | RPC]
 # config -s config.devices.total=8 (decrement this value when deleting a managed device)

To delete the above managed device:

 # config -d config.devices.device8

The following command will synchronize the live system with the new configuration:

 # config -a

14.13 Port Log

To configure serial/network port logging:

 # config -s config.eventlog.server.address='remote server ip address'
 # config -s config.eventlog.server.logfacility='facility'

'facility' can be:
Daemon
Local 0-7
Authentication
Kernel
User

Syslog
Mail
News
UUCP

config -s config.eventlog.server.logpriority='priority'

'*priority*' can be:
Info
Alert
Critical
Debug
Emergency
Error
Notice
Warning

Assume the remote log server needs a username 'name1' and password 'secret':

 # config -s config.eventlog.server.username=name1
 # config -s config.eventlog.server.password=secret

To set the remote path as '/Black Box/logs' to save logged data:

*# config -s config.eventlog.server.path=/*Black Box*/logs*
 # config -s config.eventlog.server.type=[none | syslog | nfs | cifs | usb]

If the server type is set to usb, none of the other values need to be set. The mount point for storing on a remote USB device is */var/run/portmanager/logdir*

The following command will synchronize the live system with the new configuration:
 # config -a

14.14 Alerts

You can add an email, SNMP or NAGIOS alert by following the steps below.

The general settings for all alerts

Assume this is our second alert, and we want to send alert emails to john@Black Box.com and sms's to peter@Black Box.com:

config -s config.alerts.alert2.description=MySecondAlert
config -s config.alerts.alert2.email=john@Black Box.com
config -s config.alerts.alert2.email2=peter@Black Box.com

To use NAGIOS to notify of this alert

 # config -s config.alerts.alert2.nsca.enabled=on

To use SNMP to notify of this alert

 # config -s config.alerts.alert2.snmp.enabled=on

Increment the total alerts:

config -s config.alerts.total=2

Below are the specific settings depending on the type of alert required:

Connection Alert

To trigger an alert when a user connects to serial port 5 or network host 3:

 # config -s config.alerts.alert2.host3='host name'
 # config -s config.alerts.alert2.port5=on
 # config -s config.alerts.alert2.sensor=temp
 # config -s config.alerts.alert2.signal=DSR
 # config -s config.alerts.alert2.type=login

Signal Alert

To trigger an alert when a signal changes state on port 1:

 # config -s config.alerts.alert2.port1=on
 # config -s config.alerts.alert2.sensor=temp
 # config -s config.alerts.alert2.signal=[DSR | DCD | CTS]
 # config -s config.alerts.alert2.type=signal

Pattern Match Alert

To trigger an alert if the regular expression '.*0.0% id' is found in serial port 10's character stream.

*# config -s "config.alerts.alert2.pattern=.*0.0% id"*
 # config -s config.alerts.alert2.port10=on
 # config -s config.alerts.alert2.sensor=temp
 # config -s config.alerts.alert2.signal=DSR
 # config -s config.alerts.alert2.type=pattern

UPS Power Status Alert

To trigger an alert when *myUPS* (on localhost) or *thatUPS* (on remote host *192.168.0.50*) power status changes between on line, on battery and low battery.

 # config -s config.alerts.alert2.sensor=temp
 # config -s config.alerts.alert2.signal=DSR
 # config -s config.alerts.alert2.type=ups
 # config -s config.alerts.alert2.ups1=myUPS@localhost
 # config -s config.alerts.alert2.ups2=thatUPS@192.168.0.50

Environmental and Power Sensor Alert

 # config -s config.alerts.alert2.enviro.high.critical='critical value'
 # config -s config.alerts.alert2.enviro.high.warning='warning value'
 # config -s config.alerts.alert2.enviro.hysteresis='value'
 # config -s config.alerts.alert2.enviro.low.critical='critical value'
 # config -s config.alerts.alert2.enviro.low.warning='warning value'
 # config -s config.alerts.alert2.enviro1='Enviro sensor name'
 # config -s config.alerts.alert2.outlet#='RPCname'.outlet#
'alert2.outlet#' increments sequentially with each added outlet. The second 'outlet#' refers to the specific RPC power outlets.
 # config -s config.alerts.alert2.rpc#='RPC name'
 # config -s config.alerts.alert2.sensor=[temp | humid | load | charge]
 # config -s config.alerts.alert2.signal=DSR
 # config -s config.alerts.alert2.type=enviro
 # config -s config.alerts.alert2.ups1='UPSname@hostname'

Example1: To configure a temperature sensor alert for a sensor called 'SensorInRoom42':

 # config -s config.alerts.alert2.sensor=temp

 # config -s config.alerts.alert2.enviro.high.critical=60
 # config -s config.alerts.alert2.enviro.high.warning=50
 # config -s config.alerts.alert2.enviro.hysteresis=2
 # config -s config.alerts.alert2.enviro.low.critical=5
 # config -s config.alerts.alert2.enviro.low.warning=10
 # config -s config.alerts.alert2.enviro1=SensorInRoom42
 # config -s config.alerts.alert2.signal=DSR
 # config -s config.alerts.alert2.type=enviro

Example2: To configure a load sensor alert for outlets 2 and 4 for an RPC called 'RPCInRoom20':

 # config -s config.alerts.alert2.outlet1='RPCname'.outlet2
 # config -s config.alerts.alert2.outlet2='RPCname'.outlet4
 # config -s config.alerts.alert2.enviro.high.critical=300
 # config -s config.alerts.alert2.enviro.high.warning=280
 # config -s config.alerts.alert2.enviro.hysteresis=20
 # config -s config.alerts.alert2.enviro.low.critical=50
 # config -s config.alerts.alert2.enviro.low.warning=70
 # config -s config.alerts.alert2.rpc1=RPCInRoom20
 # config -s config.alerts.alert2.sensor=load
 # config -s config.alerts.alert2.signal=DSR
 # config -s config.alerts.alert2.type=enviro

Alarm Sensor Alert

To set an alert for 'doorAlarm' and 'windowAlarm' that are two alarms connected to an environmental sensor called 'SensorInRoom3'. Both alarms are disabled on Mondays from 8:15 am to 2:30 pm:

 # config -s config.alerts.alert2.alarm1=SensorInRoom3.alarm1 (doorAlarm)
 # config -s config.alerts.alert2.alarm1=SensorInRoom3.alarm2 (windowAlarm)
 # config -s config.alerts.alert2.alarmrange.mon.from.hour=8
 # config -s config.alerts.alert2.alarmrange.mon.from.min=15
 # config -s config.alerts.alert2.alarmrange.mon.until.hour=14
 # config -s config.alerts.alert2.alarmrange.mon.until.min=30
 # config -s config.alerts.alert2.description='description'
 # config -s config.alerts.alert2.sensor=temp
 # config -s config.alerts.alert2.signal=DSR
 # config -s config.alerts.alert2.type=alarm

To enable an alarm for the entire day:

 # config -s config.alerts.alert2.alarmrange.mon.from.hour=0
 # config -s config.alerts.alert2.alarmrange.mon.from.min=0
 # config -s config.alerts.alert2.alarmrange.mon.until.hour=0
 # config -s config.alerts.alert2.alarmrange.mon.until.min=0

The following command will synchronize the live system with the new configuration:

 # config -r alerts

14.15 SMTP & SMS

To set-up an SMTP mail or SMS server with the following details:

Outgoing server address	mail.Black Box.com
Secure connection type	SSL
Sender	John@Black Box.com

Server username	john
Server password	secret
Subject line	SMTP alerts

*# config -s config.system.smtp.server=mail.*Black Box*.com*
 # config -s config.system.smtp.encryption=SSL (can also be TLS or None)
*# config -s config.system.smtp.sender=John@*Black Box*.com*
 # config -s config.system.smtp.username=john
 # config -s config.system.smtp.password=secret
 # config -s config.system.smtp.subject=SMTP alerts

To set-up an SMTP SMS server with the same details as above:

*# config -s config.system.smtp.server2=mail.*Black Box*.com*
 # config -s config.system.smtp.encryption2=SSL (can also be TLS or None)
*# config -s config.system.smtp.sender2=John@*Black Box*.com*
 # config -s config.system.smtp.username2=john
 # config -s config.system.smtp.password2=secret
 # config -s config.system.smtp.subject2=SMTP alerts

The following command will synchronize the live system with the new configuration:

config -a

14.16 SNMP

To set-up the SNMP agent on the device:

 # config -s config.system.snmp.protocol=[UDP | TCP]
 # config -s config.system.snmp.trapport='port number' (default is 162)
 # config -s config.system.snmp.address='NMS IP network address'
 # config -s config.system.snmp.commnity='community name' (v1 and v2c only)
 # config -s config.system.snmp.engineid='ID' (v3 only)
 # config -s config.system.snmp.username='username' (v3 only)
 # config -s config.system.snmp.password='password' (v3 only)
 # config -s config.system.snmp.version=[1 | 2c | 3]

The following command will synchronize the live system with the new configuration:

 # config -a

14.17 Administration

To change the administration settings to:

System Name	og.mydomain.com
System Password (root account)	secret
Description	Device in office 2

 # config -s config.system.name=og.mydomain.com
 # config -P config.system.password (will prompt user for a password)
 # config -s "config.system.location=Device in office 2"

NOTE: The -P parameter will prompt the user for a password, and encrypt it. You can encrypt the value of any config element using the -P parameter, but only encrypted user passwords and system passwords are supported. If any other element value were to be encrypted, the value will become inaccessible and will have to be reset.

The following command will synchronize the live system with the new configuration:

 # config -a

14.18 IP settings

To configure the primary network interface with static settings:

IP address	192.168.0.23
Netmask	255.255.255.0
Default gateway	192.168.0.1
DNS server 1	192.168.0.1
DNS server 2	192.168.0.2

 # config -s config.interfaces.wan.address=192.168.0.23
 # config -s config.interfaces.wan.netmask=255.255.255.0
 # config -s config.interfaces.wan.gateway=192.168.0.1
 # config -s config.interfaces.wan.dns1=192.168.0.1
 # config -s config.interfaces.wan.dns2=192.168.0.2
 # config -s config.interfaces.wan.mode=static
 # config -s config.interfaces.wan.media=[Auto | 100baseTx-FD | 100baseTx-HD | 10baseT-HD] 10baseT-FD

To enable bridging between all interfaces:

 # config -s config.system.bridge.enabled=on

To enable IPv6 for all interfaces

 # config -s config.system.ipv6.enabled=on

To configure the management LAN interface, use the same commands as above but replace:

config.interfaces.wan, with *config.interfaces.lan*

Note: Not all devices have a management LAN interface.

To configure a failover device in case of an outage:

 # config -s config.interfaces.wan.failover.address1='ip address'
 # config -s config.interfaces.wan.failover.address2='ip address'
 # config -s config.interfaces.wan.failover.interface=[eth1 | console | modem]

The network interfaces can also be configured automatically:

 # config -s config.interfaces.wan.mode=dhcp
 # config -s config.interfaces.lan.mode=dhcp

The following command will synchronize the live system with the new configuration:

 # /bin/config –-run=ipconfig

The following command will synchronize the live system with the new configuration:

 # config -r ipconfig

14.19 Date & Time Settings

To enable NTP using a server at pool.ntp.org, issue the following commands:

 # config -s config.ntp.enabled=on
 # config -s config.ntp.server=pool.ntp.org

Alternatively, you can manually change the clock settings:

To change running system time:

 # date 092216452005.05 Format is MMDDhhmm[[CC]YY][.ss]

Then the following command will save this new system time to the hardware clock:

 # /bin/hwclock -systohc

Alternatively, to change the hardware clock:

 # /bin/hwclock -- set --date=092216452005.05 Format is MMDDhhmm[[CC]YY][.ss]

Then the following command will save this new hardware clock time as the system time:

 # /bin/hwclock -hctosys

To change the timezone:

 # config -s config.system.timezone=US/Eastern

The following command will synchronize the live system with the new configuration:

 # config -r time

14.20 Dial-in settings

To enable dial-in access on the DB9 serial port from the command line with the following attributes:

Local IP Address	172.24.1.1
Remote IP Address	172.24.1.2
Authentication Type:	MSCHAPv2
Serial Port Baud Rate:	115200
Serial Port Flow Control:	Hardware
Custom Modem Initialization:	ATQ0V1H0
Callback phone	0800223665
User to dial as	user1
Password for user	secret

Run the following commands:

 # config -s config.console.ppp.localip=172.24.1.1
 # config -s config.console.ppp.remoteip=172.24.1.2
 # config -s config.console.ppp.auth=MSCHAPv2
 # config -s config.console.speed=115200
 # config -s config.console.flow=Hardware
 # config -s config.console.initstring=ATQ0V1H0
 # config -s config.console.ppp.enabled=on
 # config -s config.console.ppp.callback.enabled=on
 # config -s config.console.ppp.callback.phone1=0800223665
 # config -s config.console.ppp.username=user1
 # config -s config.console.ppp.password=secret

To make the dialed connection the default route:

 # config -s config.console.ppp.defaultroute=on

Please note that supported authentication types are 'None', 'PAP', 'CHAP' and 'MSCHAPv2'.
Supported serial port baud-rates are '9600', '19200', '38400', '57600', '115200', and '230400'.
Supported parity values are 'None', 'Odd', 'Even', 'Mark' and 'Space'.
Supported data-bits values are '8', '7', '6' and '5'.

Supported stop-bits values are '1', '1.5' and '2'.
Supported flow-control values are 'Hardware', 'Software' and 'None'.

If you do not want to use out-of-band dial-in access, note that the procedure for enabling start-up messages on the console port is covered in Chapter 15—Accessing the Console Port.

The following command will synchronize the live system with the new configuration:

 # config -a

14.21 DHCP server

To enable the DHCP server on the console management LAN, with settings:

Default lease time	200000 seconds
Maximum lease time	300000 seconds
DNS server1	192.168.2.3
DNS server2	192.168.2.4
Domain name	company.com
Default gateway	192.168.0.1
IP pool 1 start address	192.168.0.20
IP pool 1 end address	192.168.0.100
Reserved IP address	192.168.0.50
MAC to reserve IP for	00:1e:67:82:72:d9
Name to identify this host	John-PC

Issue the commands:

 # config -s config.interfaces.lan.dhcpd.enabled=on
 # config -s config.interfaces.lan.dhcpd.defaultlease=200000
 # config -s config.interfaces.lan.dhcpd.maxlease=300000
 # config -s config.interfaces.lan.dhcpd.dns1=192.168.2.3
 # config -s config.interfaces.lan.dhcpd.dns2=192.168.2.4
 # config -s config.interfaces.lan.dhcpd.domain=company.com
 # config -s config.interfaces.lan.dhcpd.gateway=192.168.0.1
 # config -s config.interfaces.lan.dhcpd.pools.pool1.start=192.168.0.20
 # config -s config.interfaces.lan.dhcpd.pools.pool1.end=192.168.0.100
 # config -s config.interfaces.lan.dhcpd.pools.total=1
 # config -s config.interfaces.lan.dhcpd.staticips.staticip1.ip=192.168.0.50
 # config -s config.interfaces.lan.dhcpd.staticips.staticip1.mac=00:1e:67:82:72:d9
 # config -s config.interfaces.lan.dhcpd.staticips.staticip1.host=John-PC
 # config -s config.interfaces.lan.dhcpd.staticips.total=1

The following command will synchronize the live system with the new configuration:

 # config -a

14.22 Services

You can manually enable or disable network servers from the command line. For example, if you wanted to guarantee the following server configuration:

HTTP Server	Enabled
HTTPS Server	Disabled
Telnet Server	Disabled
SSH Server	Enabled
SNMP Server	Disabled
Ping Replies (Respond to ICMP echo requests)	Disabled

TFTP server Enabled

 # config -s config.services.http.enabled=on
 # config -d config.services.https.enabled
 # config -d config.services.telnet.enabled
 # config -s config.services.ssh.enabled=on
 # config -d config.services.snmp.enabled
 # config -d config.services.pingreply.enabled
 # config -s config.services.tftp.enabled=on

To set secondary port ranges for any service

 # config -s config.services.telnet.portbase='port base number' Default: 2000
 # config -s config.services.ssh.portbase='port base number' Default: 3000
 # config -s config.services.tcp.portbase='port base number' Default: 4000
 # config -s config.services.rfc2217.portbase='port base number' Default: 5000
 # config -s config.services.unauthtel.portbase='port base number Default: 6000

The following command will synchronize the live system with the new configuration:

config -a

14.23 NAGIOS

To configure NAGIOS with the following settings:

NAGIOS host name	console at R3 (Name of this system)
NAGIOS host address	192.168.0.1 (IP to find this device at)
NAGIOS server address	192.168.0.10 (upstream NAGIOS server)
Enable SDT for NAGIOS ext.	Enabled
SDT gateway address	192.168.0.1 (defaults to host address)
Prefer NRPE over NSCA	Disabled (defaults to Disabled)

 # config -s config.system.nagios.enabled=on
 # config -s config.system.nagios.name=les1116
 # config -s config.system.nagios.address=192.168.0.1
 # config -s config.system.nagios.server.address=192.168.0.10
 # config -s config.system.nagios.sdt.disabled=on (diables SDT for nagios extensions)
 # config -s config.system.nagios.sdt.address=192.168.0.1
 # config -s config.system.nagios.nrpe.prefer=''

To configure NRPE with following settings:

NRPE port	5600 (port to listen on for nrpe. Defualts to 5666)
NRPE user	user1 (User to run as. Defaults to nrpe)
NRPE group	group1 (Group to run as. Defaults to nobody)
Allow command arguments	Enabled

 # config -s config.system.nagios.nrpe.enabled=on
 # config -s config.system.nagios.nrpe.port=5600
 # config -s config.system.nagios.user=user1
 # config -s config.system.nagios.nrpe.group=group1
 # config -s config.system.nagios.nrpe.cmdargs=on

To configure NSCA with the following settings:

NSCA encryption BLOWFISH (can be: [None | XOR | DES | TRPLEDES | CAST-256 | BLOWFISH | TWOFISH | RIJNDAEL-256 | SERPENT | GOST]

NSCA password	secret
NSCA check-in interval	5 minutes
NSCA port	5650 (defaults to 5667)
user to run as	User1 (defaults to nsca)
group to run as	Group1 (defaults to nobody)

 # config -s config.system.nagios.nsca.enabled=on
 # config -s config.system.nagios.nsca.encryption=BLOWFISH
 # config -s config.system.nagios.nsca.secret=secret
 # config -s config.system.nagios.nsca.interval=2
 # config -s config.system.nagios.nsca.port=5650
 # config -s config.system.nagios.nsca.user=User1
 # config -s config.system.nagios.nsca.group=Group1

Then synchronize the live system with the new configuration using *# config -a*

Chapter 15 Advanced Configuration

Introduction

Black Box *console server*s run the embedded Linux operating system. So *Administrator* class users can configure the *console server* and monitor and manage attached serial console and host devices from the command line using Linux commands and the *config* utility as described in *Chapter 14*.

The Linux kernel in the *console server* also supports GNU *bash* shell script enabling the *Administrator* to run custom scripts. This chapter presents a number of useful scripts and scripting tools including:

- ***delete-node*** which is a general script for deleting users, groups, hosts, UPSes etc.
- ***ping-detect*** which will run specified commands when a specific host stops responding to ping requests.

This chapter then details how to perform advanced and custom management tasks using Black Box commands, Linux commands, and the open source tools embedded in the *console server*:

- ***portmanager*** serial port management
- raw data access to the ports and modems
- *iptables* modifications and updating IP filtering rules
- modifying SNMP with *net-snmpd*
- public key authenticated SSH communications
- SSL, configuring HTTPS and issuing certificates
- using ***pmpower*** for *NUT* and *PowerMan* power device management
- using *IPMItools*
- CDK custom development kit

15.1 Custom Scripting

The *console server* supports GNU *bash* shell commands (refer to *Appendix A*) enabling the *Administrator* to run custom scripts.

15.1.1 Custom script to run when booting

The */etc/config/rc.local* script runs whenever the system boots. By default, this script file is empty. You can add any commands to this file if you want them to run at boot time (for example, if you wanted to display *hello world*:)

```
#!/bin/sh
echo "Hello World!"
```

If this script has been copied from a Windows machine, you may need to run the following command on the script before *bash* can run it successfully:

 # dos2unix /etc/config/rc.local

Another scenario would be to call another custom script from the */etc/config/rc.local* file, making sure that your custom script will run whenever the system is booted.

15.1.2 Running custom scripts when alerts are triggered

Whenever an alert gets triggered, specific scripts get called. These scripts all reside in */etc/scripts/*. Below is a list of the default scripts that get run for each applicable alert:

- For a connection alert (when a user connects or disconnects from a port or network host): */etc/scripts/portmanager-user-alert* (for port connections) or */etc/scripts/sdt-user-alert* (for host connections)
- For a signal alert (when a signal on a port changes state): */etc/scripts/portmanager-signal-alert*
- For a pattern match alert (when a specific regular expression is found in the serial ports character stream): */etc/scripts/portmanager-pattern-alert*
- For a UPS status alert (when the UPS power status changes between on line, on battery, and low battery): */etc/scripts/ups-status-alert*
- For a environmental, power and alarm sensor alerts (temperature, humidity, power load, and battery charge alerts): */etc/scripts/environmental-alert*
- For an interface failover alert: */etc/scripts/interface-failover-alert*

All of these scripts do a check to see whether you have created a custom script to run instead. The code that does this check is shown below (an extract from the file /etc/scripts/portmanager-pattern-alert):

```
# If there's a user-configured script, run it instead
scripts[0]="/etc/config/scripts/pattern-alert.${ALERT_PORTNAME}"
scripts[1]="/etc/config/scripts/portmanager-pattern-alert"
for (( i=0 ; i < ${#scripts[@]} ; i++ )); do
    if [ -f "${scripts[$i]}" ]; then
        exec /bin/sh "${scripts[$i]}"
    fi
done
```

This code shows that there are two alternative scripts that can be run instead of the default one. This code first checks whether a file "*/etc/config/scripts/pattern-alert.${ALERT_PORTNAME}*" exists. The variable ${ALERT_PORTNAME} must be replaced with "port01" or "port13" or whichever port the alert should run for. If this file cannot be found, the script checks whether the file "*/etc/config/scripts/portmanager-pattern-alert*" exists. If either of these files exists, the script calls the *exec* command on the first file that it finds and runs that custom file/script instead.

As an example, you can copy the */etc/scripts/portmanager-pattern-alert* script file to */etc/config/scripts/portmanager-pattern-alert*:

```
# cd /
# mkdir /etc/config/scripts   (if the directory does not already exist)
# cp /etc/scripts/portmanager-pattern-alert /etc/config/scripts/portmanager-pattern-alert
```

The next step will be to edit the new script file. First, open the file */etc/config/scripts/portmanager-pattern-alert* using vi (or any other editor), and remove the lines that check for a custom script (the code from above). This will prevent the new custom script from repeatedly calling itself. After these lines have been removed, edit the file, or add any additional scripting to the file.

15.1.3 Example script - Power Cycling on Pattern Match

For example, we have an RPC (PDU) connected to port 1 on a *console server* and also have some telecommunications device connected to port 2 (which is powered by the RPC outlet 3). Now assume the telecom device transmits a character stream "EMERGENCY" out on its serial console port every time that it encounters some specific error, and the only way to fix this error is to power cycle the telecom device.

The first step is to setup a pattern-match alert on port 2 to check for the pattern "EMERGENCY".

Next we need to create a custom script to deal with this alert:

 # cd /
 # mkdir /etc/config/scripts (if the directory does not already exist)
 # cp /etc/scripts/portmanager-pattern-alert /etc/config/scripts/portmanager-pattern-alert

Note: Make sure to remove the *if* statement (which checks for a custom script) from the new script, in order to prevent an infinite loop.

The *pmpower* utility is used to send power commands to RPC device in order to power cycle our telecom device:

 # pmpower -l port01 -o 3 cycle (The RPC is on serial port 1. The telecom device is powered by RPC outlet 3)

We can now append this command to our custom script. This will guarantee that our telecom device will be power cycled every time the console reads the "EMERGENCY" character stream on port 2.

15.1.4 Example script - Multiple email notifications on each alert

If you want to send more than one email when an alert triggers, you have to create a replacement script using the method described above and add the appropriate lines to your new script.

Currently, there is a script */etc/scripts/alert-email* that runs from within all the alert scripts (for example, *portmanager-user-alert or environmental-alert*). The alert-email script sends the email. The line that invokes the email script is as follows:

/bin/sh /etc/scripts/alert-email $suffix &

If you want to send another email to a single address or the same email to many recipients, edit the custom script appropriately. You can follow the examples in any of the seven alert scripts listed above. In particular, consider the *portmanager-user-alert* script. If you need to send the same alert email to more than one email address, find the lines in the script responsible for invoking the alert-email script, then add the following lines below the existing lines:

export TOADDR="emailaddress@domain.com"
/bin/sh /etc/scripts/alert-email $suffix &

These two lines assign a new email address to TOADDR and invoke the alert-email script in the background.

15.1.5 Deleting Configuration Values from the CLI

The *delete-node* script is provided to help with deleting nodes from the command line. The *"delete-node"* script takes one argument, the node name you want to delete (for example, *"config.users.user1"* or *"config.sdt.hosts.host1"*).

delete-node is a general script for deleting any node you desire (users, groups, hosts, UPSes, etc.) from the command line. The script deletes the specified node and shuffles the remainder of the node values.

For example, if we have five users configured and we use the script to delete user 3, then user 4 will become user 3, and user 5 will become user 4.

This creates an obvious complication because this script does NOT check for any other dependencies that the node being deleted may have. You are responsible for making sure that any references and dependencies connected to the deleted node are removed or corrected in the config.xml file.

The script treats all nodes the same. The syntax to run the script is *# ./delete-node {node name}.* To remove user 3:

```
# ./delete-node config.users.user3
```

The *delete-node* script

```
#!/bin/bash
#User must provide the node to be removed. e.g. "config.users.user1"
# Usage: delete-node {full node path}

if [ $# != 1 ]
then
        echo "Wrong number of arguments"
        echo "Usage: delnode {full '.' delimited node path}"
        exit 2
fi

# test for spaces
TEMP=`echo "$1" | sed 's/.* .*/N/'`
if [ "$TEMP" = "N" ]
then
        echo "Wrong input format"
        echo "Usage: delnode {full '.' delimited node path}"
        exit 2
fi

# testing if node exists
TEMP=`config -g config | grep "$1"`
if [ -z "$TEMP" ]
then
        echo "Node $1 not found"
        exit 0
fi

# LASTFIELD is the last field in the node path e.g. "user1"
# ROOTNODE is the upper level of the node e.g. "config.users"
# NUMBER is the integer value extracted from LASTFIELD e.g. "1"
# TOTALNODE is the node name for the total e.g. "config.users.total"
# TOTAL is the value of the total number of items before deleting e.g. "3"
# NEWTOTAL is the modified total i.e. TOTAL-1
# CHECKTOTAL checks if TOTAL is the actual total items in .xml

LASTFIELD=${1##*.}
ROOTNODE=${1%.*}
```

```
NUMBER=`echo $LASTFIELD | sed 's/^[a-zA-Z]*//g'`
TOTALNODE=`echo ${1%.*} | sed 's/\(.*\)/\1.total/'`
TOTAL=`config -g $TOTALNODE | sed 's/.* //'`
NEWTOTAL=$[ $TOTAL -1 ]

# Make backup copy of config file
cp /etc/config/config.xml /etc/config/config.bak
echo "backup of /etc/config/config.xml saved in /etc/config/config.bak"


if [ -z $NUMBER ] # test whether a singular node is being \
#deleted e.g. config.sdt.hosts
then

        echo "deleting $1"
        config -d "$1"

        echo Done
        exit 0

elif [ $NUMBER = $TOTAL ] # Test if only one item exists
then
        echo "only one item exists"
        # Deleting node
        echo "Deleting $1"
        config -d "$1"

        # Modifying item total.
        config -s "$TOTALNODE=0"

        echo Done
        exit 0

elif [ $NUMBER -lt $TOTAL ] # more than one item exists
then

        # Modify the users list so user numbers are sequential
        # by shifting the users into the gap one at a time...

        echo "Deleting $1"

        LASTFIELDTEXT=`echo $LASTFIELD | sed 's/[0-9]//g'`
        CHECKTOTAL=`config -g $ROOTNODE.$LASTFIELDTEXT$TOTAL`

        if [ -z "$CHECKTOTAL" ]
        then
                echo "WARNING: "$TOTALNODE" greater than number of items"
        fi

        COUNTER=1
        while [ $COUNTER != $((TOTAL-NUMBER+1)) ]
        do
```

```
                config -g $ROOTNODE.$LASTFIELDTEXT$((NUMBER+COUNTER)) \
                | while read LINE
                do
                        config -s \
                        "`echo "$LINE" | sed -e "s/$LASTFIELDTEXT$((NUMBER+ \
                        COUNTER))/$LASTFIELDTEXT$((NUMBER+COUNTER-1))/" \
                        -e 's/ /=/'`"

                done

                let COUNTER++
        done

        # deleting last user
        config -d $ROOTNODE.$LASTFIELDTEXT$TOTAL

        # Modifying item total.
        config -s "$TOTALNODE=$NEWTOTAL"

        echo Done
        exit 0
else
        echo "error: item being deleted has an index greater than total items. Increase the total count
variable."
        exit 0
fi
```

15.1.6 Power Cycle any device when a ping request fails

The *ping-detect* script is designed to run specified commands when a monitored host stops responding to ping requests.

The first parameter taken by the *ping-detect* script is the hostname/IP address of the device to ping. Any other parameters are then regarded as a command to run whenever the ping to the host fails. *ping-detect* can run any number of commands.

Below is an example using *ping-detect* to power cycle an RPC (PDU) outlet whenever a specific host fails to respond to a ping request. The *ping-detect* runs from */etc/config/rc.local* to make sure that the monitoring starts whenever the system boots.

Suppose we have a serially controlled RPC connected to port01 on a *console server* and have a router powered by outlet 3 on the RPC (and the router has an internal IP address of 192.168.22.2). The following instructions will show you how to continuously ping the router. When the router fails to respond to a series of pings, the *console server* will send a command to RPC outlet 3 to power cycle the router, and write the current date/time to a file:

- Copy the *ping-detect* script to */etc/config/scripts/* on the *console server*
- Open */etc/config/rc.local* using vi
- Add the following line to *rc.local*:

  /etc/config/scripts/ping-detect 192.168.22.2 /bin/bash -c "pmpower -l port01 -o 3 cycle && date" > /tmp/output.log &

The above command will cause the *ping-detect* script to continuously ping the host at 192.168.22.2 which is the router. If the router crashes, it will no longer respond to ping requests. If this happens, the two commands *pmpower* and *date* will run. The output from these commands is sent to the file */tmp/output.log* so that we have a record. The *ping-detect* is also run in the background using the *"&"*.

Remember the *rc.local* script only runs by default when the system boots. You can manually run the *rc.local* script or the *ping-detect* script if desired.

The *ping-detect* script

The above is just one example of using the *ping-detect* script. The idea of the script is to run any number of commands when a specific host stops responding to ping requests. Here are details of the *ping-detect* script itself:

```
#!/bin/sh
# Usage: ping-detect HOST [COMMANDS...]
# This script takes 2 types of arguments: hostname/IPaddress to ping, and the commands to
#  run if the ping fails 5 times in a row. This script can only take one host/IPaddress per
# instance. Multiple independent commands can be sent to the script. The commands will be
# run one after the other.
#
# PINGREP is the entire reply from the ping command
# LOSS is the percentage loss from the ping command
# $1 must be the hostname/IPaddress of device to ping
# $2... must be the commands to run when the pings fail.
COUNTER=0
TARGET="$1"
shift
# loop indefinitely:
while true
do
        # ping the device 10 times
        PINGREP=`ping -c 10 -i 1 "$TARGET" `
        #get the packet loss percentage
        LOSS=`echo "$PINGREP" | grep "%" | sed -e 's/.* \([0-9]*\)% .*/\1/'`
        if [ "$LOSS" -eq "100" ]
        then
                COUNTER=`expr $COUNTER + 1`
        else
                COUNTER=0
                sleep 30s
        fi
        if [ "$COUNTER" -eq 5 ]
        then
                COUNTER=0
                "$@"
                sleep 2s
        fi
done
```

15.1.7 Running custom scripts when a configurator is invoked

A configurator is responsible for reading the values in */etc/config/config.xml* and making the appropriate changes live. Some changes made by the configurators are part of the Linux configuration itself, such as user passwords or *ipconfig*.

Currently there are nineteen configurators. Each one is responsible for a specific group of config (for example, the *"users"* configurator makes the user configurations in the *config.xml file* live). To see all the available configurators type the following from a command line prompt:

 # config

When a change is made using the Management Console web GUI, the appropriate configurator automatically runs. This can be a problem if another *Administrator* makes a change using the Management Console. The configurator could possibly overwrite any custom CLI/linux configurations you may have set.

The solution is to create a custom script that runs after each configurator runs. After each configurator runs, it will check whether that appropriate custom script exists. You can then add any commands to the custom script and they will be invoked after the configurator runs.

The custom scripts must be in the correct location:

/etc/config/scripts/config-post-

To create an alerts custom script:

 # cd /etc/config/scripts
 # touch config-post-alerts
 # vi config-post-alerts

You could use this script to recover a specific backup config or overwrite a config or make copies of config files, etc.

15.1.8 Backing-up the configuration and restoring using a local USB stick

The ***/etc/scripts/backup-usb*** script is written to save and load custom configuration using a USB flash disk. Before saving configuration locally, you must prepare the USB storage device for use. To do this, disconnect all USB storage devices except for the storage device you want to use.

Usage: */etc/scripts/backup-usb* COMMAND [FILE]

COMMAND:

check-magic -- check volume label
set-magic -- set volume label
save [FILE] -- save configuration to USB
delete [FILE] -- delete a configuration tarbal from USB
list -- list available config backups on USB
load [FILE] -- load a specific config from USB
load-default -- load the default configuration
set-default [FILE] -- set which file becomes the default

The first thing to do is to check if the USB disk has a label:

 # /etc/scripts/backup-usb check-magic

If this command returns "Magic volume not found", then run the following command:

 # /etc/scripts/backup-usb set-magic

To save the configuration:

 # /etc/scripts/backup-usb save config-20May

To check if the backup was saved correctly:

 # /etc/scripts/backup-usb list

If this command does not display *"* config-20May"* then there was an error saving the configuration.

The set-default command takes an input file as an argument and renames it to "default.opg". This default configuration remains stored on the USB disk. The next time you want to load the default config, it will be sourced from the new default.opg file. To set a config file as the default:

 # /etc/scripts/backup-usb set-default config-20May

To load this default:

 # /etc/scripts/backup-usb load-default

To load any other config file:

 # /etc/scripts/backup-usb load {filename}

The */etc/scripts/backup-usb* script can be executed directly with various *COMMANDS* or called from other custom scripts you may create. We recommend that you do not customize the */etc/scripts/backup-usb* script itself at all.

15.1.9 Backing-up the configuration off-box

If you do not have a USB port on your *console server,* you can back up the configuration to an off-box file. Before backing up you need to arrange a way to transfer the backup off-box. This could be via an NFS share, a Samba (Windows) share to USB storage, or copied off-box via the network. If backing up directly to off-box storage, make sure it is mounted.

/tmp is not a good location for the backup except as a temporary location before transferring it off-box. The */tmp* directory will not survive a reboot. The */etc/config* directory is not a good place either, because it will not survive a restore.

Backup and restore should be done by the root user to make sure correct file permissions are set. The config command is used to create a backup tarball:

config -e <Output File>

The tarball will be saved to the indicated location. It will contain the contents of the */etc/config/* directory in an uncompressed and unencrypted form.

Example nfs storage:

 # mount -t nfs 192.168.0.2:/backups /mnt # config -e /mnt/les4108.config # umount/mnt/

Example transfer off-box via scp:

 # config -e /tmp/les4108.config
 # scp /tmp/les4108.config 192.168.0.2:/backups

The config command is also used to restore a backup:

config -i <Input File>

This will extract the contents of the previously created backup to */tmp*, and then synchronize the */etc/config* directory with the copy in */tmp*.

One problem that can crop up here is that there is not enough room in */tmp* to extract files to. The following command will temporarily increase the size of */tmp*:

mount -t tmpfs -o remount,size=2048k tmpfs /var

If restoring to either a new unit or one that has been factory defaulted, make sure that the process generating SSH keys either stops or completes before restoring configuration. If this is not done, then a mix of old and new keys may be put in place.

SSH uses these keys to avoid man-in-the-middle attacks. Logging in may be disrupted.

15.2 Advanced Portmanager

Black Box's *portmanger* program manages the *console server* serial ports. It routes network connection to serial ports, checks permissions, and monitors and logs all the data flowing to/from the ports.

15.2.1 Portmanager commands

pmshell

The *pmshell* command acts similar to the standard *tip* or *cu* commands, but all serial port access is directed *via* the portmanager.

Example: To connect to port 8 *via* the portmanager:

 # pmshell -l port08

pmshell Commands:

Once connected, the pmshell command supports a subset of the '~' escape commands that tip/cu support. For SSH you must prefix the escape with an additional '~' command (i.e. use the '~~' escape)

Send Break: Typing the character sequence '~b' will generate a BREAK on the serial port.

History: Typing the character sequence '~h' will generate a history on the serial port.

Quit pmshell: Typing the character sequence '~.' will exit from pmshell.

Set RTS to 1 run the command: *pmshell --rts=1*

Show all signals: *# pmshell –signals*

DSR=1 DTR=1 CTS=1 RTS=1 DCD=0

Read a line of text from the serial port: *# pmshell –getline*

pmchat

The *pmchat* command acts similar to the standard *chat* command, but all serial port access is directed *via* the portmanager.

Example: To run a chat script *via* the portmanager:

 # pmchat -v -f /etc/config/scripts/port08.chat < /dev/port08

For more information on using *chat* (and *pmchat*) you should consult the UNIX man pages:

http://techpubs.sgi.com/library/tpl/cgibin/getdoc.cgi?coll=linux&db=man&fname=/usr/share/catman/man8/chat.8.html

pmusers

The *pmusers* command is used to query the portmanager for active user sessions.

Example: To detect which users are currently active on which serial ports:

 # pmusers

This command will output nothing if there are no active users currently connected to any ports. Otherwise, it will respond with a sorted list of usernames per active port:

Port 1:
user1
user2
Port 2:
user1
Port 8:
user2

The above output indicates that a user named "*user1*" is actively connected to ports 1 and 2, while "*user2*" is connected to both ports 1 and 8.

portmanager daemon

There is normally no need to stop and restart the daemon. To restart the daemon normally, just run the command:

 # portmanager

Supported command line options are:

Force portmanager to run in the foreground: *--nodaemon*

Set the level of debug logging: *--loglevel={debug,info,warn,error,alert}*

Change which configuration file it uses: *-c /etc/config/portmanager.conf*

Signals

Sending a SIGHUP signal to the portmanager will cause it to re-read its configuration file

15.2.2 External Scripts and Alerts

The portmanager can execute external scripts on certain events.

When the portmanager opens a port:

- It attempts to execute */etc/config/scripts/portXX.init* (where XX is the number of the port, *e.g.* 08). The script is run with STDIN and STDOUT both connected to the serial port.
- If the script cannot be executed, then portmanager will execute */etc/config/scripts/portXX.chat via* the chat command on the serial port.

When an alert occurs on a port:

- The portmanager will attempt to execute */etc/config/scripts/portXX.alert* (where XX is the port number, e.g. 08)
- The script is run with STDIN containing the data which triggered the alert, and STDOUT redirected to /dev/null, NOT to the serial port. If you want to communicate with the port, use *pmshell* or *pmchat* from within the script.
- If the script cannot be executed, then the alert will be mailed to the address configured in the system administration section.

When a user connects to any port:

- If a file called */etc/config/pmshell-start.sh* exists it is run when a user connects to a port. It is provided 2 arguments, the "Port number" and the "Username". Here is a simple example:

```
</etc/config/pmshell-start.sh >
#!/bin/sh
PORT="$1"
USER="$2"
echo "Welcome to port $PORT $USER"
< /etc/config/pmshell-start.sh>
```

- The return value from the script controls whether the user is accepted or not, if 0 is returned (or nothing is done on exit as in the above script) the user is permitted, otherwise the user is denied access.
- Here is a more complex script which reads from configuration to display the port label if available and denies access to the root user:

```
</etc/config/pmshell-start.sh>
#!/bin/sh
PORT="$1"
USER="$2"
LABEL=$(config -g config.ports.port$PORT.label | cut -f2- -d' ')
if [ "$USER" == "root" ]; then
        echo "Permission denied for Super User"
        exit 1
fi
if [  -z  "$LABEL" ]; then
echo "Welcome $USER, you are connected to Port $PORT"
else
echo "Welcome $USER, you are connected to Port $PORT ($LABEL)"
fi
</etc/config/pmshell-start.sh>
```

15.3 Raw Access to Serial Ports

15.3.1 Access to serial ports

You can use *tip* and *stty* to completely bypass the *portmanager* and have raw access to the serial ports.

When you run *tip* on a *portmanager* controlled port, *portmanager* closes that port, and stops monitoring it until *tip* releases control of it.

With *stty*, the changes made to the port only "stick" until that port is closed and opened again. People probably will not want to use *stty* for more than initial debugging of the serial connection.

If you want to use *stty* to configure the port, you can put *stty* commands in */etc/config/scripts/portXX.init* which gets run whenever portmanager opens the port.

Otherwise, any setup you do with *stty* will get lost when the portmanager opens the port. (The reason that portmanager sets things back to its *config* rather than using whatever is on the port, is so the port is in a known good state, and will work, no matter what things are done to the serial port outside of portmanager.)

15.3.2 Accessing the console/modem port

The console dial-in is handled by *mgetty*, with automatic PPP login extensions. *mgetty* is a smart *getty* replacement, designed to be used with Hayes compatible data and data/fax modems. *mgetty* knows about modem initialization, manual modem answering (your modem doesn't answer if the machine isn't ready), UUCP locking (you can use the same device for dial-in and dial-out). *mgetty* provides very extensive logging facilities. All standard *mgetty* options are supported.

Modem initialization strings:

- To override the standard modem initialization string either use the Management Console (refer *Chapter 5*) or the command line config tool (refer to *Dial-In Configuration Chapter 14*).

Enabling Boot Messages on the Console:

- If you are not using a modem on the DB9 console port and instead want to connect to it directly via a Null Modem cable, enable verbose mode, which allows you to see the standard linux start-up messages. Follow these commands:

  # /bin/config --set=config.console.debug=on # /bin/config --run=console # reboot

- If at some point in the future you chose to connect a modem for dial-in out-of-band access, you can reverse the procedure with the following commands.

  # /bin/config --del=config.console.debug # /bin/config --run=console # reboot

15.4 IP- Filtering

The *console server* uses the *iptables* utility to provide a stateful firewall of LAN traffic. By default, rules are automatically inserted to allow access to enabled services, and serial port access *via* enabled protocols. The commands that add these rules are contained in configuration files:

/etc/config/ipfilter

This is an executable shell script that runs whenever the LAN interface is brought up and whenever modifications are made to the *iptables* configuration as a result of CGI actions or the *config* command line tool.

The basic steps performed are as follows:

- The current *iptables* configuration is erased.
- If a customized IP-Filter script exists it is executed and no other actions are performed.
- Standard policies are inserted that will drop all traffic not explicitly allowed to and through the

system.

- Rules are added which explicitly allow network traffic to access enabled services*,* for example, TTP, SNMP*, etc.*
- Rules are added that explicitly allow traffic network traffic access to serial ports over enabled protocols *e.g.* Telnet, SSH and raw TCP.

If the standard system firewall configuration is not adequate for your needs you can bypass it safely by creating a file at **/etc/config/filter-custom** containing commands to build a specialized firewall. This firewall script will run whenever the LAN interface is brought up (including initially) and will override any automated system firewall settings.

Below is a simple example of a custom script that creates a firewall using the *iptables* command. Only incoming connections from computers on a C-class network 192.168.10.0 will be accepted when this script is installed at */etc/config/filter-custom.* Note that when this script is called, any preexisting chains and rules have been flushed from *iptables*:

```
#/bin/sh
# Set default policies to drop any incoming or routable traffic
# and blindly accept anything from the 192.168.10.0 network.
iptables –-policy FORWARD DROP
iptables –-policy INPUT DROP
iptables –-policy OUTPUT ACCEPT
# Allow responses to outbound connections back in.
iptables –-append INPUT \
        –-match state –-state ESTABLISHED,RELATED –-jump ACCEPT
# Explicitly accept any connections from computers on
# 192.168.10.0/24
iptables –-append INPUT –-source 192.168.10.0/24 –-jump ACCEPT
```

There's good documentation about using the *iptables* command at the Linux *netfilter* website *http://netfilter.org/documentation/index.html*. There are also many high-quality tutorials and HOWTOs available *via* the *netfilter* website, in particular peruse the tutorials listed on the *netfilter* HOWTO page.

15.5 Modifying SNMP Configuration

15.5.1 */etc/config/snmpd.conf*

The *net-snmpd* is an extensible SNMP agent which responds to SNMP queries for management information from SNMP management software. Upon receiving a request, it processes the request(s), collects the requested information and/or performs the requested operation(s) and returns the information to the sender.

This includes built-in support for a wide range of MIB information modules, and can be extended using dynamically loaded modules, external scripts and commands. *snmpd* when enabled should run with a default configuration. You can customize its behavior via the options in */etc/config/snmpd.conf.*
To change standard system information such as system contact, name, and location, edit */etc/config/snmpd.conf* file and locate the following lines:

```
sysdescr        "Black Box"
syscontact      root <root@localhost>(configure /etc/default/snmpd.conf)
```

sysname	*Not defined (edit /etc/default/snmpd.conf)*
syslocation	*Not defined (edit /etc/default/snmpd.conf)*

Simply change the values of *sysdescr, syscontact, sysname* and *syslocation* to the desired settings and restart *snmpd*.

The *snmpd.conf* provides is extremely powerful and too flexible to completely cover here. The configuration file itself is commented extensively and good documentation is available at the *net-snmp* website *http://www.net-snmp.org*, specifically:

Man Page:	*http://www.net-snmp.org/docs/man/snmpd.conf.html*
FAQ:	*http://www.net-snmp.org/docs/FAQ.html*
Net-SNMPD Tutorial:	*http://www.net-snmp.org/tutorial/tutorial-5/demon/snmpd.html*

15.5.2 Adding more than one SNMP server

To add more than one SNMP server for alert traps add the first SNMP server using the Management Console (refer Chapter 7) or the command line *config* tool. Secondary and any further SNMP servers are added manually using *config.*

Log in to the *console server*'s command line shell as root or an admin user. Refer back to the Management Console UI or user documentation for descriptions of each field.

To set the Manager Protocol field:
config --set config.system.snmp.protocol2=UDP or
config --set config.system.snmp.protocol2=TCP

To set the Manager Address field:
config --set config.system.snmp.address2=w.x.y.z
.. replacing *w.x.y.z* with the IP address or DNS name.

To set the Manager Trap Port field
config --set config.system.snmp.trapport2=162
.. replacing 162 with the TCP/UDP port number

To set the Version field
config --set config.system.snmp.version2=1 or
config --set config.system.snmp.version2=2c or
config --set config.system.snmp.version2=3

To set the Community field (SNMP version 1 and 2c only)
config --set config.system.snmp.community2=yourcommunityname
.. replacing *yourcommunityname* with the community name

To set the Engine ID field (SNMP version 3 only)
config --set config.system.snmp.engineid2=800000020109840301
.. replacing *800000020109840301* with the engine ID

To set the Username field (SNMP version 3 only)
config --set config.system.snmp.username2=yourusername

.. replacing *yourusername* with the username
config.system.snmp.username2 (3 only)

To set the Engine ID field (SNMP version 3 only)
config --set config.system.snmp.password2=yourpassword
.. replacing *yourpassword* with the password

Once the fields are set, apply the configuration with the following command:
config --run snmp

You can add a third or more SNMP servers by incrementing the "2" in the above commands, e.g. config.system.snmp.protocol3, config.system.snmp.address3, etc.

15.6 Secure Shell (SSH) Public Key Authentication

This section covers how to generate public and private keys in a Linux and Windows environment and configure SSH for public key authentication. The steps to use in a Clustering environment are:

- Generate a new public and private key pair.
- Upload the keys to the Master and to each Slave *console server.*
- Fingerprint each connection to validate.

15.6.1 SSH Overview

Popular TCP/IP applications such as telnet, rlogin, ftp, and others transmit their passwords unencrypted. Doing this across pubic networks like the Internet can have catastrophic consequences. It leaves the door open for eavesdropping, connection hijacking, and other network-level attacks.

Secure Shell (SSH) is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels.

OpenSSH, the de facto open source SSH application, encrypts all traffic (including passwords) to effectively eliminate these risks. Additionally, OpenSSH provides a myriad of secure tunneling capabilities, as well as a variety of authentication methods.

OpenSSH is the port of OpenBSD's excellent OpenSSH[0] to Linux and other versions of Unix. OpenSSH is based on the last free version of Tatu Ylonen's sample implementation with all patent-encumbered algorithms removed (to external libraries), all known security bugs fixed, new features reintroduced, and many other clean-ups. http://www.openssh.com/ The only changes in the Black Box SSH implementation are:

- PAM support
- EGD[1]/PRNGD[2] support and replacements for OpenBSD library functions that are absent from other versions of UNIX
- The config files are now in */etc/config. e.g.*
  - */etc/config/sshd_config* instead of */etc/sshd_config*
  - */etc/config/ssh_config* instead of */etc/ssh_config*
  - */etc/config/users/<username>/.ssh/* instead of */home/<username>/.ssh/*

15.6.2 Generating Public Keys (Linux)

To generate new SSH key pairs use the Linux *ssh-keygen* command. This will produce an RSA or DSA public/private key pair and you will be prompted for a path to store the two key files, for example, *id_dsa.pub* (the public key) and *id_dsa* (the private key). For example:

$ ssh-keygen -t [rsa|dsa]
Generating public/private [rsa|dsa] key pair.
Enter file in which to save the key *(/home/user/.ssh/id_[rsa|dsa]):*
Enter *passphrase* (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in */home/user/.ssh/id_[rsa|dsa].*
Your public key has been saved in */home/user/.ssh/id_[rsa|dsa].pub.*
The key fingerprint is:
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
$

Create a new directory to store your generated keys. You can also name the files after the device they will be used for. For example:

$ mkdir keys
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa): */home/user/keys/control_room*
Enter *passphrase* (empty for no passphrase):
Enter same *passphrase* again:
Your identification has been saved in */home/user/keys/control_room*
Your public key has been saved in */home/user/keys/control_room.pub*.
The key fingerprint is:
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
$

Make sure that there is no password associated with the keys. If there is a password, then the Black Box devices will have no way to supply it as runtime.

Full documentation for the *ssh-keygen* command can be found at *http://www.openbsd.org/cgi-bin/man.cgi?query=ssh-keygen*

15.6.3 Installing the SSH Public/Private Keys (Clustering)

For Black Box *console server*s, the keys can be simply uploaded through the web interface, on the **System: Administration** page. This enables you to upload stored RSA or DSA Public Key pairs to the Master and apply the Authorized key to the slave and is described in Chapter 4. Once complete, you then proceed to Fingerprinting as described below.

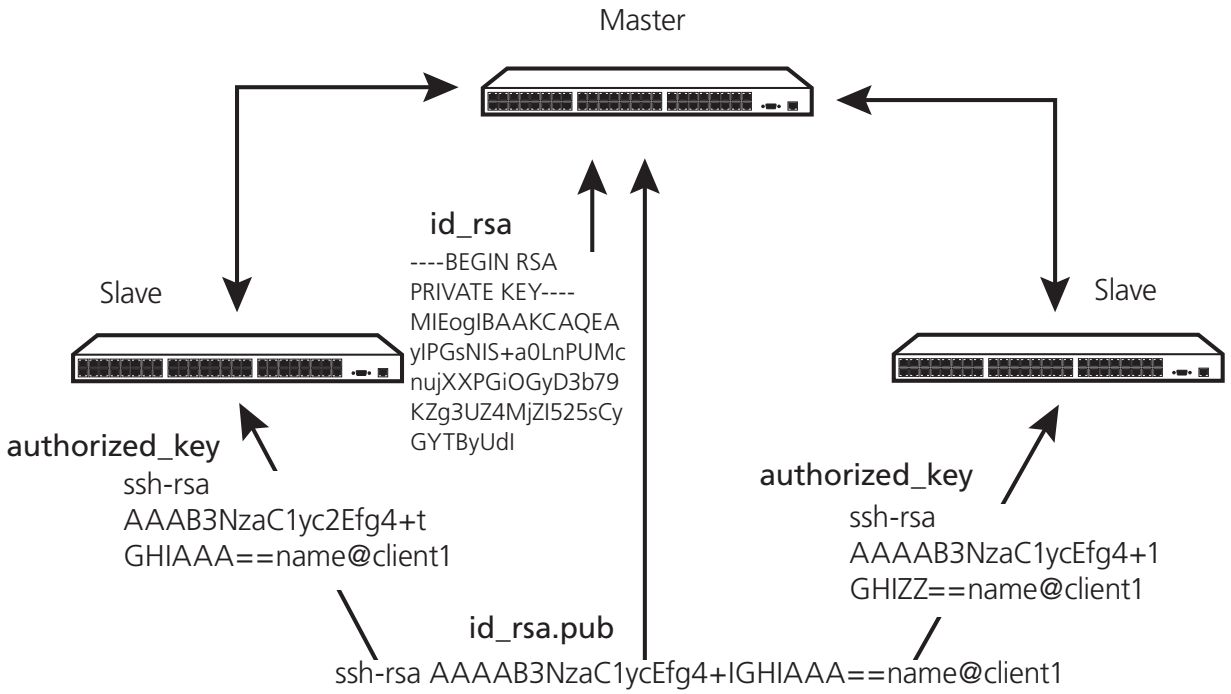
15.6.4 Installing SSH Public Key Authentication (Linux)

Alternately, the public key can be installed on the unit remotely from the linux host with the *scp* utility as follows.
Assuming the user on the Management Console is called "fred"; the IP address of the *console server* is 192.168.0.1 (default); and the public key is on the *linux/unix* computer in *~/.ssh/id_dsa.pub*. Execute the following command on the *linux/unix* computer:

*scp ~/.ssh/id_dsa.pub *
root@192.168.0.1:/etc/config/users/fred/.ssh/authorized_keys

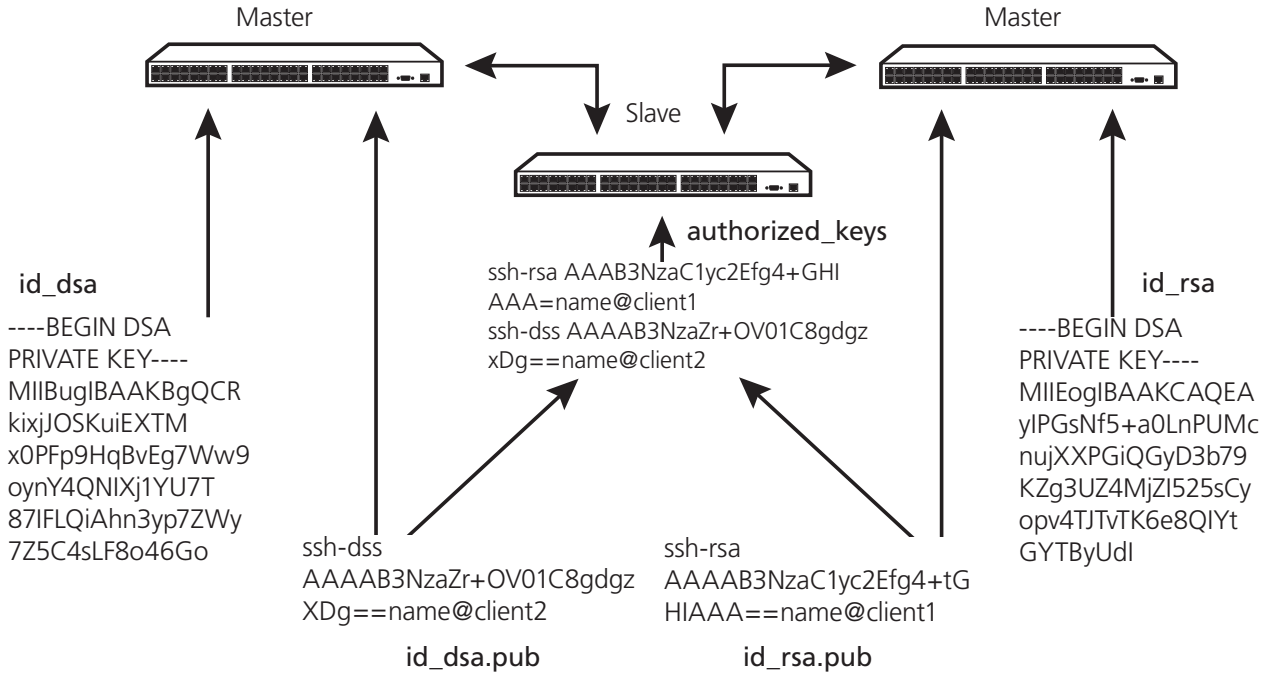
The authorized_keys file on the *console server* needs to be owned by "fred", so login to the Management Console as **root** and type:

chown fred /etc/config/users/fred/.ssh/authorized_keys



If the Black Box device selected to be the server will only have one client device, then the

authorized_keys file is simply a copy of the public key for that device. If one or more devices will be clients of the server, then the *authorized_keys* file will contain a copy of all of the public keys. RSA and DSA keys may be freely mixed in the authorized_keys file. For example, assume we already have one server, called bridge_server, and two sets of keys, for the control_room and the plant_entrance:
$ ls /home/user/keys control_room control_room.pub plant_entrance plant_entrance.pub $ cat /home/user/keys/control_room.pub /home/user/keys/plant_entrance.pub > /home/user/keys/authorized_keys_bridge_server



More documentation on OpenSSH can be found at:

http://openssh.org/portable.html
http://www.openbsd.org/cgi-bin/man.cgi?query=ssh&sektion=1
http://www openbsd.org/cgi-bin/man.cgi?query=sshd.

15.6.5 Generating public/private keys for SSH (Windows)

This section describes how to generate and configure SSH keys using Windows.

First create a new user from the Black Box Management (the following example uses a user called "testuser") making sure it is a member of the "users" group.

If you do not already have a public/private key pair you can generate them now using *ssh-keygen*, *PuTTYgen* or a similar tool:

PuTTYgen: *http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html*

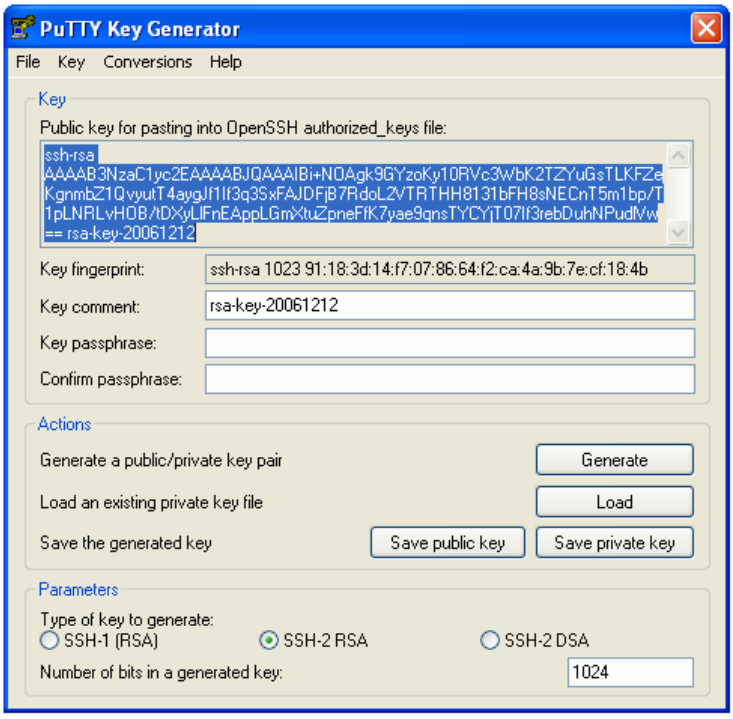
OpenSSH: *http://www.openssh.org/*

OpenSSH (Windows): *http://sshwindows.sourceforge.net/download/*

For example, using PuTTYgen, make sure you have a recent version of the *puttygen.exe* (available from *http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)* Make sure you have a recent version of WinSCP (available from *http://winscp.net/eng/download.php*)

To generate a SSH key using PuTTY *http://sourceforge.net/docs/F02/#clients:*

- Execute the PUTTYGEN.EXE program.
- Select the desired key type *SSH2 DSA* (you may use RSA or DSA) within the *Parameters* section.
- It is important that you leave the passphrase field blank.
- Click on the *Generate* button.
- Follow the instruction to move the mouse over the blank area of the program in order to create random data used by PUTTYGEN to generate secure keys. Key generation will occur once PUTTYGEN has collected sufficient random data.



- Create a new file " *authorized_keys* " (with notepad) and copy your public key data from the "Public key for pasting into OpenSSH authorized_keys file" section of the PuTTY Key Generator, and paste the key data to the "authorized_keys" file. Make sure there is only one line of text in this file.
- Use WinSCP to copy this "authorized_keys" file into the users home directory: e.g. */etc/config/users/testuser/.ssh/authorized_keys* of the Black Box gateway which will be the SSH server. You will need to make sure this file is in the correct format with the correct permissions with the following commands:

*# dos2unix *
*/etc/config/users/testuser/.ssh/authorized_keys && chown testuser *
/etc/config/users/testuser/.ssh/authorized_keys

- Using WinSCP copy the attached sshd_config over */etc/config/sshd_config* on the server (Makes sure public key authentication is enabled).
- Test the Public Key by logging in as "testuser" Test the Public Key by logging in as "testuser" to the client Black Box device and typing (you should not need to enter anything): # ssh -o StrictHostKeyChecking=no <server-ip>

To automate connection of the SSH tunnel from the client on every power-up you need to make the *clients /etc/config/rc.local* look like the following:

#!/bin/sh
ssh -L9001:127.0.0.1:4001 -N -o StrictHostKeyChecking=no testuser@<server-ip> &

This will run the tunnel redirecting local port 9001 to the server port 4001.

15.6.6 Fingerprinting

Fingerprints are used to ensure you are establishing an SSH session to who you think you are. On the first connection to a remote server you will receive a fingerprint that you can use on future connections.

This fingerprint is related to the host key of the remote server. Fingerprints are stored in *~/.ssh/known_hosts*.

To receive the fingerprint from the remote server, log in to the client as the required user (usually root) and establish a connection to the remote host:

 # ssh remhost
The authenticity of host 'remhost (192.168.0.1)' can't be established.
RSA key fingerprint is 8d:11:e0:7e:8a:6f:ad:f1:94:0f:93:fc:7c:e6:ef:56.
Are you sure you want to continue connecting (yes/no)?

At this stage, answer yes to accept the key. You should get the following message:

Warning: Permanently added 'remhost,192.168.0.1' (RSA) to the list of known hosts.

You may be prompted for a password, but there is no need to log in— you have received the fingerprint and can Ctrl-C to cancel the connection. If the host key changes you will receive the following warning, and not be allowed to connect to the remote host:

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@ IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Someone could be eavesdropping on you right now (man-in-the-middle attack)!

It is also possible that the RSA host key has just been changed.

The fingerprint for the RSA key sent by the remote host is

ab:7e:33:bd:85:50:5a:43:0b:e0:bd:43:3f:1c:a5:f8.

Please contact your system *Administrator*.

Add correct host key in */.ssh/known_hosts* to get rid of this message.

Offending key in */.ssh/known_hosts:1*

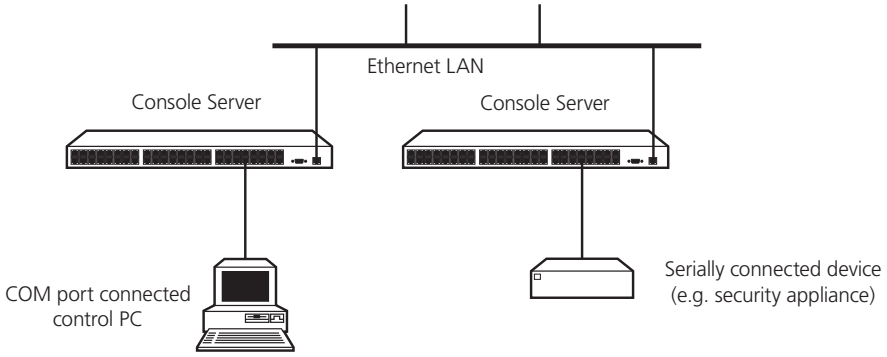
RSA host key for *remhost* has changed and you have requested strict checking.

Host key verification failed.

If the host key has been legitimately changed, it can be removed from the *~/.ssh/known_hosts* file and the new fingerprint added. If it has not changed, this indicates a serious problem that should be investigated immediately.

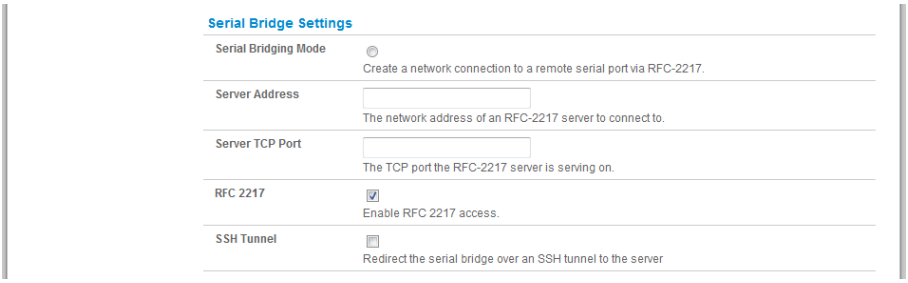
15.6.7 SSH tunneled serial bridging

You have the option to apply SSH tunneling when two Black Box console servers are configured for serial bridging.



As detailed in *Chapter 4*, the *Server* console server is setup in *Console server* mode with either RAW or RFC2217 enabled and the *Client* console server is set up in Serial Bridging Mode with the Server Address, and Server TCP Port (4000 + port for RAW or 5000 + port # for RFC2217) specified:

- Select **SSH Tunnel** when configuring the **Serial Bridging Setting.**


Serial Bridge Settings
Serial Bridging Mode
Create a network connection to a remote serial port via RFC-2217.
Server Address
The network address of an RFC-2217 server to connect to.
Server TCP Port
The TCP port the RFC-2217 server is serving on.
RFC 2217
Enable RFC 2217 access.
SSH Tunnel
Redirect the serial bridge over an SSH tunnel to the server

Next, you will need to set up SSH keys for each end of the tunnel and upload these keys to the *Server* and *Client* console servers.

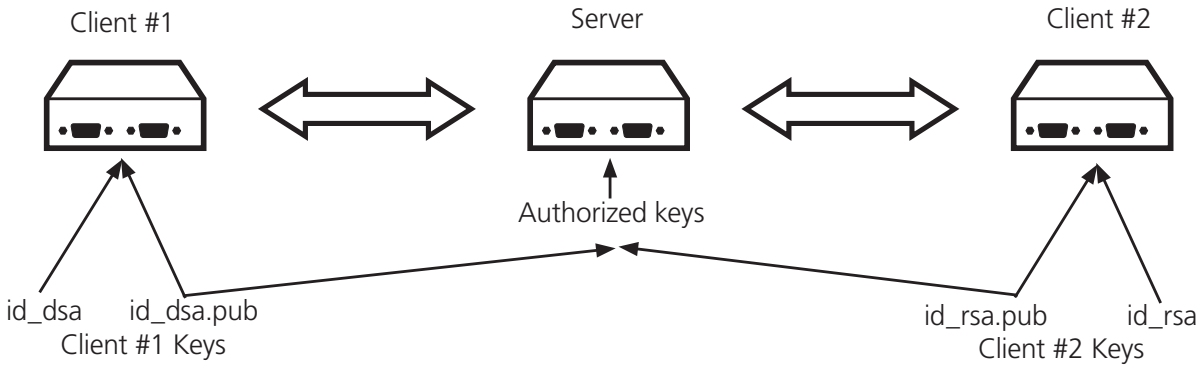
Client Keys:

The first step in setting up ssh tunnels is to generate keys. Ideally, you will use a separate, secure, machine to generate and store all keys to be used on the *console servers*. If this is not ideal for your situation, keys may be generated on the *console servers* themselves.

It is possible to generate only one set of keys, and reuse them for every SSH session. While we do not recommend this, each organization will need to balance the security of separate keys against the additional administration they bring.

Generated keys may be one of two types—RSA or DSA (and it is beyond the scope of this document to recommend one over the other). RSA keys will go into the files *id_rsa* and *id_rsa.pub*. DSA keys will be stored in the files *id_dsa* and *id_dsa.pub*.

For simplicity going forward, the term *private key* will be used to refer to either *id_rsa* or *id_dsa* and *public key* to refer to either *id_rsa.pub* or *id_dsa.pub.*



To generate the keys using OpenBSD's OpenSSH suite, we use the *ssh-keygen* program:

$ ssh-keygen -t [rsa|dsa]
Generating public/private [rsa|dsa] key pair.
Enter file in which to save the key *(/home/user/.ssh/id_[rsa|dsa]):*
Enter *passphrase* (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in */home/user/.ssh/id_[rsa|dsa].*
Your public key has been saved in */home/user/.ssh/id_[rsa|dsa].pub.*
The key fingerprint is:
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
$

It is advisable to create a new directory to store your generated keys. It is also possible to name the files after the device they will be used for. For example:

$ mkdir keys
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa): */home/user/keys/control_room*
Enter *passphrase* (empty for no passphrase):
Enter same *passphrase* again:
Your identification has been saved in */home/user/keys/control_room*
Your public key has been saved in */home/user/keys/control_room.pub*.
The key fingerprint is:
28:aa:29:38:ba:40:f4:11:5e:3f:d4:fa:e5:36:14:d6 user@server
$

You should ensure there is no password associated with the keys. If there is a password, then the *console servers* will have no way to supply it as runtime.

Authorized Keys:

If the *console server* selected to be the server will only have one client device, then the *authorized_keys* file is simply a copy of the public key for that device. If one or more devices will be clients of the server, then the *authorized_keys* file will contain a copy of all of the public keys. RSA and DSA keys may be freely mixed in the *authorized_keys* file.

For example, assume we already have one server, called *bridge_server*, and two sets of keys, for the *control_room* and the *plant_entrance*:

$ ls /home/user/keys

control_room control_room.pub plant_entrance plant_entrance.pub

$ cat /home/user/keys/control_room.pub

/home/user/keys/plant_entrance.pub >

/home/user/keys/authorized_keys_bridge_server

Uploading Keys:

The keys for the server can be uploaded through the web interface, on the **System: Administration** page as detailed earlier. If only one client will be connecting, then simply upload the appropriate public key as the authorized keys file. Otherwise, upload the authorized keys file constructed in the previous step.

Each client will then need its own set of keys uploaded through the same page. Take care to ensure that the correct type of keys (DSA or RSA) go in the correct spots, and that the public and private keys are in the correct spot.

15.6.8 SDT Connector Public Key Authentication

SDT Connector can authenticate against a *console servers* using your SSH key pair, rather than requiring you to enter your password (i.e. public key authentication).

- To use public key authentication with SDT Connector, you must first create an RSA or DSA key pair (using *ssh-keygen, PuTTYgen* or a similar tool) and add the public part of your SSH key pair to the Black Box gateway—as described in the earlier section.
- Next, add the private part of your SSH key pair (this file is typically named *id_rsa* or *id_dsa*) to SDT Connector client. Click **Edit -> Preferences -> Private Keys -> Add**, locate the private key file and click **OK**. You do not have to add the public part of your SSH key pair, it is calculated using the private key.

SDT Connector will now use public key authentication when SSH connecting through the *console server*. You may have to restart SDT Connector to shut down any existing tunnels that were established using password authentication.

If you have a host behind the *console server* that you connect to by clicking the SSH button in SDT Connector, you can also configure it for public key authentication. Essentially what you are using is SSH over SSH, and the two SSH connections are entirely separate, and the host configuration is entirely independent of SDT Connector and the *console server*. You must configure the SSH client that SDT Connector launches (e.g. Putty, OpenSSH) and the host's SSH server for public key authentication.

15.7 Secure Sockets Layer (SSL) Support

Secure Sockets Layer (SSL) is a protocol developed by Netscape for transmitting private documents *via* the Internet. SSL works by using a private key to encrypt data that's transferred over the SSL connection.

The *console server* includes OpenSSL. The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation.

OpenSSL is based on the excellent SSLeay library developed by Eric A. Young and Tim J. Hudson. The OpenSSL toolkit is licensed under an Apache-style licence, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions. In the *console server,* OpenSSL is used primarily in conjunction with 'http' to have secure browser access to the GUI management console across insecure networks.

More documentation on OpenSSL is available from:

http://www.openssl.org/docs/apps/openssl.html

http://www.openssl.org/docs/HOWTO/certificates.txt

15.8 HTTPS

The Management Console can be served using HTTPS by running the webserver *via sslwrap*. The server can be launched on request using *inetd*.

The HTTP server provided is a slightly modified version of the *fnord-httpd* from *http://www.fefe.de/fnord/*

The SSL implementation is provided by the *sslwrap* application compiled with OpenSSL support. You can find more detailed documentation at *http://www.rickk.com/sslwrap/*

If your default network address is changed or the unit is to be accessed *via* a known Domain Name, you can use the following steps to replace the default SSL Certificate and Private Key with ones tailored for your new address.

15.8.1 Generating an encryption key

To create a 1024 bit RSA key with a password, issue the following command on the command line of a linux host with the *openssl* utility installed:

openssl genrsa -des3 -out ssl_key.pem 1024

15.8.2 Generating a self-signed certificate with OpenSSL

This example shows how to use OpenSSL to create a self-signed certificate. OpenSSL is available for most Linux distributions *via* the default package management mechanism. (Windows users can check *http://www.openssl.org/related/binaries.html*)

To create a 1024 bit RSA key and a self-signed certificate, issue the following *openssl* command from the host you have *openssl* installed on:

openssl req -x509 -nodes -days 1000 \

-newkey rsa:1024 -keyout ssl_key.pem -out ssl_cert.pem

You will be prompted to enter a lot of information. Most of it doesn't matter, but the "Common Name" should be the domain name of your computer (*e.g.* test.Black Box.com). When you have entered everything, the certificate will be created in a file called *ssl_cert.pem*.

15.8.3 Installing the key and certificate

We recommend that you use an SCP (Secure Copying Protocol) client to copy files securely to the *console server* unit. The *scp* utility is distributed with OpenSSH for most Unix distributions, while Windows users can use something like the PSCP command line utility available with PuTTY.

You can install remotely the files created in the steps above with the *scp* utility as follows:

scp ssl_key.pem root@<address of unit>:/etc/config/
scp ssl_cert.pem root@<address of unit>:/etc/config/

or using PSCP:

pscp -scp ssl_key.pem root@<address of unit>:/etc/config/
pscp -scp ssl_cert.pem root@<address of unit>:/etc/config/

PuTTY and the PSCP utility can be downloaded from:
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

More detailed documentation on the PSCP can be found:
http://the.earth.li/~sgtatham/putty/0.58/htmldoc/Chapter5.html#pscp

15.8.4 Launching the HTTPS Server

Note that the easiest way to enable the HTTPS server is from the web Management Console. Simply click the appropriate checkbox in **Network -> Services -> HTTPS Server** and the HTTPS server will be activated (assuming the *ssl_key.pem* & *ssl_cert.pem* files exist in the */etc/config* directory).

Alternatively *inetd* can be configured to launch the secure *fnord* server from the command line of the unit as follows.

Edit the *inetd* configuration file. From the unit command line:

vi /etc/config/inetd.conf

Append a line:

443 stream tcp nowait root sslwrap -cert /etc/config/ssl_cert.pem -key /etc/config/ssl_key.pem -exec /bin/httpd /home/httpd"

Save the file and signal *inetd* of the configuration change.

kill -HUP `cat /var/run/inetd.pid`

The HTTPS server should be accessible from a web client at a URL similar to this: *https://<common name of unit>*

More detailed documentation about the *openssl* utility can be found at the website:
http://www.openssl.org/

15.9 Power Strip Control

The *console server* supports a growing list of remote power-control devices (RPCs) that you can configure using the Management Console as described in Chapter 8. These RPCs are controlled using the open source *PowerMan* and *Network UPS Tools* and with Black Box's *pmpower* utility.

15.9.1 The PowerMan tool

PowerMan provides power management in a data center or compute cluster environment. It performs operations such as power on, power off, and power cycle via remote power controller (RPC) devices.

Synopsis
powerman [-option] [targets]
pm [-option] [targets]

Options
-1, --on Power ON targets.
-0, --off Power OFF targets.
-c, --cycle Power cycle targets.
-r, --reset Assert hardware reset for targets (if implemented by RPC).
-f, --flash Turn beacon ON for targets (if implemented by RPC).
-u, --unflash Turn beacon OFF for targets (if implemented by RPC).
-l, --list List available targets. If possible, output will be compressed into a host range (see TARGET SPECIFICATION below).
-q, --query Query plug status of targets. If none specified, query all targets. Status is not cached; each time this option is used, powermand queries the appropriate RPC's. Targets connected to RPC's that could not be contacted (e.g. due to network failure) are reported as status "unknown". If possible, output will be compressed into host ranges.
-n, --node Query node power status of targets (if implemented by RPC). If no targets specified, query all targets. In this context, a node in the OFF state could be ON at the plug but operating in standby power mode.
-b, --beacon Query beacon status (if implemented by RPC). If no targets are specified, query all targets.
-t, --temp Query node temperature (if implemented by RPC). If no targets are specified, query all targets. Temperature information is not interpreted by powerman and is reported as received from the RPC on one line per target, prefixed by target name.
-h, --help Display option summary.
-L, --license Show powerman license information.
-d, --destination host[:port] Connect to a powerman daemon on non-default host and optionally port.
-V, --version Display the powerman version number and exit.
-D, --device Displays RPC status information. If targets are specified, only RPC's matching the target list are displayed.
-T, --telemetry Causes RPC telemetry information to be displayed as commands are processed. Useful for debugging device scripts.
-x, --exprange Expand host ranges in query responses.

For more details refer *http://linux.die.net/man/1/powerman*

Also refer *powermand* (*http://linux.die.net/man/1/powermand)* documentation and *powerman.conf* (*http://linux.die.net/man/5/powerman.conf*)

Target Specification
powerman target hostnames may be specified as comma separated or space separated hostnames or host ranges. Host ranges are of the general form: prefix[n-m,l-k,...], where n < m and l < k, etc., This form should not be confused with regular expression character classes (also denoted by ''[]''). For example, foo[19] does not represent foo1 or foo9, but rather represents a degenerate range: foo19.

This range syntax is meant only as a convenience on clusters with a prefix NN naming convention and specification of ranges should not be considered necessary—the list foo1,foo9 could be specified as such, or by the range foo[1,9].

Some examples of powerman targets follows.

Power on hosts bar,baz,foo01,foo02,...,foo05: *powerman --on bar baz foo[01-05]*

Power on hosts bar,foo7,foo9,foo10: *powerman --on bar,foo[7,9-10]*

Power on foo0,foo4,foo5: *powerman --on foo[0,4-5]*

As a reminder to the reader, some shells will interpret brackets ([and]) for pattern matching. Depending on your shell, you might need to enclose ranged lists within quotes. For example, in tcsh, the last example above should be executed as:

powerman --on "foo[0,4-5]"

15.9.2 The *pmpower* tool

The *pmpower* utility is a high level tool for manipulating remote preconfigured power devices connected to the *console server* either via a serial or network connection. The PDU UPS and IPMI power devices are variously controlled using the open source *PowerMan, IPMItool* or *Network UPS Tools* and Black Box's *pmpower* utility arches over these tools so the devices can be controlled through one command line:

pmpower [-?h] [-l device | -r host] [-o outlet] [-u username] [-p password] action

-?/-h This help message.
-l The serial port to use.
-o The outlet on the power target to apply to
-r The remote host address for the power target
-u Override the configured username
-p Override the configured password
on This *action* switches the specified device or outlet(s) on
off This *action* switches the specified device or outlet(s) off
cycle This *action* switches the specified device or outlet(s) off and on again
status This *action* retrieves the current status of the device or outlet

Examples:

To turn outlet 4 of the power device connected to serial port 2 on: *# pmpower -l port02 -o 4 on*

To turn an IPMI device off located at IP address 192.168.1.100 (where username is 'root' and password is 'calvin': *# pmpower -r 192.168.1.100 -u root -p calvin off*

Default system Power Device actions are specified in */etc/powerstrips.xml*. Custom Power Devices can be added in */etc/config/powerstrips.xml*. If an action is attempted which has not been configured for a specific Power Device, *pmpower* will exit with an error.

15.9.3 Adding new RPC devices

There are a number of simple paths to adding support for new RPC devices.

The first is to have scripts to support the particular RPC included in either the open source *PowerMan* project *(http://sourceforge.net/projects/powerman)* or the open source *NUT UPS Tools* project. The *PowerMan* device specifications are rather weird and it is suggested that you leave the actual writing of these scripts to the PowerMan authors. Documentation on how they work can be found at *http://linux.die.net/man/5/powerman.dev*. The *Network UPS Tools (NUT)* project has recently moved on from its UPS management origins to also cover SNMP PDUs (and embrace PowerMan). Black Box progressively includes the updated *PowerMan* and *NUT* build into the *console server* firmware releases.

The second path is to directly add support for the new RPC devices (or to customize the existing RPC device support) on your particular *console server*. The **Manage: Power** page uses information contained in */etc/powerstrips.xml* to configure and control devices attached to a serial port. The configuration also looks for (and loads) */etc/config/powerstrips.xml* if it exists.

The user can add their own support for more devices by putting definitions for them into */etc/config/powerstrips.xml*. This file can be created on a host system and copied to the Management Console device using *scp*. Alternatively, login to the Management Console and use *ftp* or *wget* to transfer files.

Here is a brief description of the elements of the XML entries in */etc/config/powerstrips.xml*.

```
<powerstrip>
        <id>Name or ID of the device support</id>
        <outlet port="port-id-1">Display Port 1 in menu</outlet>
        <outlet port="port-id-2">Display Port 2 in menu</outlet>
        ...
        <on>script to turn power on</on>
        <off>script to power off</off>
        <cycle>script to cycle power</cycle>
        <status>script to write power status to /var/run/power-status</status>
        <speed>baud rate</speed>
        <charsize>character size</charsize>
        <stop>stop bits</stop>
        <parity>parity setting</parity>
</powerstrip>
```

The *id* appears on the web page in the list of available devices types to configure.

The outlets describe targets that the scripts can control. For example, a power control board may control several different outlets. The port-id is the native name for identifying the outlet. This value will be passed to the scripts in the environment variable *outlet*, allowing the script to address the correct outlet.

There are four possible scripts: *on, off, cycle* and *status*.

When a script is run, its standard input and output is redirected to the appropriate serial port. The script receives the outlet and port in the *outlet* and *port* environment variables respectively.

The script can be anything that can be executed within the shell.

All of the existing scripts in */etc/powerstrips.xml* use the *pmchat* utility.

pmchat works just like the standard unix "chat" program, only it ensures interoperation with the port manager.

The final options, *speed, charsize, stop* and *parity* define the recommended or default settings for the attached device.

15.10 IPMItool

The *console server* includes the *ipmitool* utility for managing and configuring devices that support the Intelligent Platform Management Interface (IPMI) version 1.5 and version 2.0 specifications.

IPMI is an open standard for monitoring, logging, recovery, inventory, and control of hardware that is implemented independent of the main CPU, BIOS, and OS. The service processor (or Baseboard Management Controller, BMC) is the brain behind platform management and its primary purpose is to handle the autonomous sensor monitoring and event logging features.

The *ipmitool* program provides a simple command-line interface to this BMC. It features the ability to read the sensor data repository (SDR) and print sensor values, display the contents of the System Event Log (SEL), print Field Replaceable Unit (FRU) inventory information, read and set LAN configuration parameters, and perform remote chassis power control.

SYNOPSIS

ipmitool [**-c**|**-h**|**-v**|**-V**] **-I** *open* <*command*>

ipmitool [**-c**|**-h**|**-v**|**-V**] **-I** *lan* **-H** <*hostname*>
[**-p** <*port*>]
[**-U** <*username*>]
[**-A** <*authtype*>]
[**-L** <*privlvl*>]
[**-a**|**-E**|**-P**|**-f** <*password*>]
[**-o** <*oemtype*>]
<*command*>

ipmitool [**-c**|**-h**|**-v**|**-V**] **-I** *lanplus* **-H** <*hostname*>
[**-p** <*port*>]
[**-U** <*username*>]
[**-L** <*privlvl*>]
[**-a**|**-E**|**-P**|**-f** <*password*>]
[**-o** <*oemtype*>]
[**-C** <*ciphersuite*>]
<*command*>

DESCRIPTION

This program lets you manage Intelligent Platform Management Interface (IPMI) functions of either the local system, via a kernel device driver, or a remote system, using IPMI V1.5 and IPMI v2.0. These functions include printing FRU information, LAN configuration, sensor readings, and remote chassis power control.

IPMI management of a local system interface requires a compatible IPMI kernel driver to be installed and configured. On Linux, this driver is called *OpenIPMI* and it is included in standard distributions. On Solaris, this driver is called *BMC* and is inclued in Solaris 10. Management of a remote station requires the IPMI-over-LAN interface to be enabled and configured. Depending on the particular requirements of each system, it may be possible to enable the LAN interface using *ipmitool* over the system interface.

OPTIONS

-a Prompt for the remote server password.

-A <*authtype*>
Specify an authentication type to use during IPMIv1.5 *lan* session activation. Supported types are NONE, PASSWORD, MD5, or OEM.

-c Present output in CSV (comma separated variable) format. This is not available with all commands.

-C <*ciphersuite*>
The remote server authentication, integrity, and encryption algorithms to use for IPMIv2 *lanplus* connections. See table 22-19 in the IPMIv2 specification. The default is 3 which specifies RAKP-HMAC-SHA1 authentication, HMAC-SHA1-96 integrity, and AES-CBC-128 encryption algorightms.

-E The remote server password is specified by the environment variable *IPMI_PASSWORD*.

-f <*password_file*>
Specifies a file containing the remote server password. If this option is absent, or if password_file is empty, the password will default to NULL.

-h Get basic usage help from the command line.

-H <*address*>
Remote server address, can be IP address or hostname. This option is required for *lan* and *lanplus* interfaces.

-I <*interface*>
Selects IPMI interface to use. Supported interfaces that are compiled in are visible in the usage help output.

-L <*privlvl*>
Force session privilege level. Can be CALLBACK, USER, OPERATOR, ADMIN. Default is ADMIN.

-m <*local_address*>
Set the local IPMB address. The default is 0x20 and there should be no need to change it for normal operation.

-o <*oemtype*>
Select OEM type to support. This usually involves minor hacks in place in the code to work around quirks in various BMCs from various manufacturers. Use *-o list* to see a list of current supported OEM types.

-p <*port*>
Remote server UDP port to connect to. Default is 623.

-P <*password*>
Remote server password is specified on the command line. If supported it will be obscured in the process list. **Note!** Specifying the password as a command line option is not recommended.

-t <*target_address*>
Bridge IPMI requests to the remote target address.

-U <*username*>
Remote server username, default is NULL user.

-v Increase verbose output level. This option may be specified multiple times to increase the level of debug output. If given three times you will get hexdumps of all incoming and outgoing packets.

-V Display version information.

If no password method is specified, then *ipmitool* will prompt the user for a password. If no password is entered at the prompt, the remote server password will default to NULL.

SECURITY

The *ipmitool* documentation highlights that there are several security issues to be considered before enabling the IPMI LAN interface. A remote station has the ability to control a system's power state as well as being able to gather certain platform information. To reduce vulnerability, we strongly advise that the IPMI LAN interface only be enabled in 'trusted' environments where system security is not an issue or where there is a dedicated secure 'management network' or access has been provided through an *console server*.

Further, we strongly advise that you do not enable IPMI for remote access without setting a password, and that that password should not be the same as any other password on that system.

When an IPMI password is changed on a remote machine with the IPMIv1.5 *lan* interface, the new password is sent across the network as clear text. This could be observed and then used to attack the remote system. We recommend that IPMI password management only be done over IPMIv2.0 *lanplus* interface or the system interface on the local station.

For IPMI v1.5, the maximum password length is 16 characters. Passwords longer than 16 characters will be truncated.

For IPMI v2.0, the maximum password length is 20 characters; longer passwords are truncated.

COMMANDS

help

This can be used to get command-line help on *ipmitool* commands. It may also be placed at the end of commands to get option usage help.

ipmitool help

Commands:

raw Send a RAW IPMI request and print response
lan Configure LAN Channels
chassis Get chassis status and set power state
event Send pre-defined events to MC
mc Management Controller status and global enables
sdr Print Sensor Data Repository entries and readings
sensor Print detailed sensor information
fru Print built-in FRU and scan SDR

for FRU locators
sel Print System Event Log (SEL)
pef Configure Platform Event Filtering (PEF)
sol Configure IPMIv2.0 Serial-over-LAN
isol Configure IPMIv1.5 Serial-over-LAN
user Configure Management Controller users
channel Configure Management Controller channels
session Print session information
exec Run list of commands from file
set Set runtime variable for shell and exec

ipmitool chassis help

Chassis Commands: status, power, identify, policy, restart_cause, poh, bootdev

ipmitool chassis power help

chassis power Commands: status, on, off, cycle, reset, diag, soft

You will find more details on *ipmitools* at http://*ipmitool*.sourceforge.net/manpage.html

15.11 Custom Development Kit (CDK)

As detailed in this manual customers can copy scripts, binaries, and configuration files directly to the *console server*.

Black Box also freely provides a development kit that allows changes to be made to the software in *console server* firmware image. The customer can use the CDK to:

- generate a firmware image without certain programs, such as telnet, which may be banned by company policy.
- generate an image with new programs, such as custom Nagios plug-in binaries or company specific binary utilities.
- generate an image with custom defaults e.g. it may be required that the *console server* be configured to have a specific default serial port profile which is reverted to even in event of a factory reset.
- place configuration files into the firmware image, which cannot then be modified e.g. # /bin/config –-set= tools update the configuration files in /etc/config which are read/write, whereas the files in /etc are read only and cannot be modified

The CDK essentially provides a snapshot of the Black Box build process (taken after the programs have been compiled and copied to a temporary directory *romfs*) just before the compressed file systems are generated. You can obtain a copy of the Black Box CDK for the particular appliance you are working with from Black Box

Note The CDK is free.

15.12 Scripts for Managing Slaves

When the *console server*s are cascaded the Master is in control of the serial ports on the Slaves, and the Master's Management Console provides a consolidated view of the settings for its own and all the Slave's serial ports. The Master does not provide a fully consolidated view, for example, *Status: Active Users* only displays those users active on the Master's ports and you will need to write a custom bash script that parses the port logs if you want to find out who's logged in to cascaded serial ports from the master.

You will probably also want to enable remote or USB logging, because local logs only buffer 8K of data and don't persist between reboots.

This script would, for example, parse each port log file line by line, each time it sees *'LOGIN: username'*, it adds username to the list of connected users for that port, each time it sees *'LOGOUT: username'* it removes it from the list. The list can then be nicely formatted and displayed. You can run the script on the remote log server. To enable log storage and connection logging:

- Select *Alerts & Logging: Port Log*
- *Configure* log storage
- Select *Serial & Network: Serial Port*, *Edit* the serial port(s)
- Under *Console server*, select *Logging Level 1* and click Apply

There's a useful tutorial on creating a bash script CGI at
http://www.yolinux.com/TUTORIALS/LinuxTutorialCgiShellScript.html

Similarly, the Master does maintain a view of the status of the slaves:
- Select *Status: Support Report*
- Scroll down to *Processes*
- Look for: */bin/ssh -MN -o ControlPath=/var/run/cascade/%h slavename*
- These are the slaves that are connected
- Note the end of the Slaves' names will be truncated, so the first 5 characters must be unique

Alternatively, you can write a custom CGI script as described above. The currently connected Slaves can be determined by running: *ls /var/run/cascade* and the configured slaves can be displayed by running: *config -g config.cascade.slaves*

Appendix A Linux Commands & Source Code

The *console server* platform is a dedicated Linux computer, optimized to provide monitoring and secure access to serial and network consoles of critical server systems and their supporting power and networking infrastructure.

Black Box *console server*s are built on the 2.4 uCLinux kernel as developed by the uCLinux project. This is GPL code and source can be found at *http://cvs.uclinux.org*. Some uCLinux commands have config files that can be altered (e.g. *portmanager, inetd*, *init*, *ssh/sshd/scp/sshkeygen, ucd-snmpd, samba, fnord, sslwrap).* Other commands you can run and do neat stuff with (e.g. *loopback*, *bash (shell)*, *ftp, hwclock*, *iproute*, *iptables*, *netcat*, *ifconfig*, *mii-tool*, *netstat*, *route*, *ping*, *portmap*, *pppd*, *routed*, *setserial*, *smtpclient*, *stty*, *stunel*, *tcpdump*, *tftp*, *tip*, *traceroute)*

Below are most of the standard uCLinux and BusyBox commands (and some custom Black Box commands) that are in the default build tree. The *Administrator* can use these to configure the *console server*, and monitor and manage attached serial console and host devices:

addgroup *	Add a group or add an user to a group
adduser *	Add an user
agetty	alternative Linux getty
arp	Manipulate the system ARP cache
arping	Send ARP requests/replies
bash	GNU Bourne-Again Shell
busybox	Swiss army knife of embedded Linux commands
cat *	Concatenate FILE(s) and print them to stdout
chat	Useful for interacting with a modem connected to stdin/stdout
chgrp *	Change file access permissions
chmod *	Change file access permissions
chown *	Change file owner and group
config	Black Box tool to manipulate and query the system configuration from the command line
cp *	Copy files and directories
date *	Print or set the system date and time
dd *	Convert and copy a file
deluser *	Delete USER from the system
df *	Report filesystem disk space usage
dhcpd	Dynamic Host Configuration Protocol server
discard	Network utility that listens on the discard port
dmesg *	Print or control the kernel ring buffer
echo *	Print the specified ARGs to stdout
erase	Tool for erasing MTD partitions
eraseall	Tool for erasing entire MTD partitions
false *	Do nothing, unsuccessful
find	Search for files

flashw	Write data to individual flash devices
flatfsd	Daemon to save RAM file systems back to FLASH
ftp	Internet file transfer program
gen-keys	SSH key generation program
getopt *	Parses command options
gettyd	Getty daemon
grep *	Print lines matching a pattern
gunzip *	Compress or expand files
gzip *	Compress or expand files
hd	ASCII, decimal, hexadecimal, octal dump
hostname *	Get or set hostname or DNS domain name
httpd	Listen for incoming HTTP requests
hwclock	Query and set hardware clock (RTC)
inetd	Network super-server daemon
inetd-echo	Network echo utility
init	Process control initialization
ip	Show or manipulate routing, devices, policy routing and tunnels
ipmitool	Linux IPMI manager
iptables	Administration tool for IPv4 packet filtering and NAT
ip6tables	Administration tool for IPv6 packet filtering
iptables-restore	Restore IP Tables
iptables-save	Save IP Tables
kill *	Send a signal to a process to end gracefully
ln *	Make links between files
login	Begin session on the system
loopback	Black Box loopback diagnostic command
loopback1	Black Box loopback diagnostic command
loopback2	Black Box loopback diagnostic command
loopback8	Black Box loopback diagnostic command
loopback16	Black Box loopback diagnostic command
loopback48	Black Box loopback diagnostic command
ls *	List directory contents
mail	Send and receive mail
mkdir *	Make directories
mkfs.jffs2	Create an MS-DOS file system under Linux
mknod *	Make block or character special files
more *	File perusal filter for crt viewing
mount *	Mount a file system
msmtp	SMTP mail client
mv *	Move (rename) files
nc	TCP/IP Swiss army knife
netflash	Upgrade firmware on ucLinux platforms using the blkmem interface
netstat	Print network connections, routing tables, interface statistics etc
ntpd	Network Time Protocol (NTP) daemon

pgrep	Display process(es) selected by regex pattern
pidof	Find the process ID of a running program
ping	Send ICMP ECHO_REQUEST packets to network hosts
ping6	IPv6 ping
pkill	Sends a signal to process(es) selected by regex pattern
pmchat	Black Box command similar to the standard chat command (via portmanager)
pmdeny	
pminetd	
pmloggerd	
pmshell	Black Box command similar to the standard *tip* or *cu* but all serial port access is directed via the portmanager.
pmusers	Black Box command to query portmanager for active user sessions
portmanager	Black Box command that handles all serial port access
portmap	DARPA port to RPC program number mapper
pppd	Point-to-Point protocol daemon
ps *	Report a snapshot of the current processes
pwd *	Print name of current/working directory
reboot *	*Soft* reboot
rm *	Remove files or directories
rmdir *	Remove empty directories
routed	Show or manipulate the IP routing table
routed	Show or manipulate the IP routing table
routef	IP Route tool to flush IPv4 routes
routel	IP Route tool to list routes
rtacct	Applet printing /proc/net/rt_acct
rtmon	RTnetlink listener
scp	Secure copy (remote file copy program)
sed *	Text stream editor
setmac	Sets the MAC address
setserial	Sets and reports serial port configuration
sh	Shell
showmac	Shows MAC address
sleep *	Delay for a specified amount of time
smbmnt	Helper utility for mounting SMB file systems
smbmount	Mount an SMBFS file system
smbumount	SMBFS umount for normal users
snmpd	SNMP daemon
snmptrap	Sends an SNMP notification to a manager
sredird	RFC 2217 compliant serial port redirector
ssh	OpenSSH SSH client (remote login program)
ssh-keygen	Authentication key generation, management, and conversion
sshd	OpenSSH SSH daemon
sslwrap	Program that allows plain services to be accessed via SSL
stty	Change and print terminal line settings
stunnel	Universal SSL tunnel

sync *	Flush file system buffers
sysctl	Configure kernel parameters at runtime
syslogd	System logging utility
tar *	The tar archiving utility
tc	Show traffic control settings
tcpdump	Dump traffic on a network
telnetd	Telnet protocol server
tftp	Client to transfer a file from/to tftp server
tftpd	Trivial file Transfer Protocol (tftp) server
tip	Simple terminal emulator/cu program for connecting to modems and serial devices
top	Provide a view of process activity in real time
touch *	Change file timestamps
traceroute	Print the route packets take to network host
traceroute6	Traceroute for IPv6
true *	Returns an exit code of TRUE (0)
umount *	Unmount file systems
uname *	Print system information
usleep *	Delay for a specified amount of time
vconfig *	Create and remove virtual ethernet devices
vi *	Busybox clone of the VI text editor
w	Show who is logged on and what they are doing
zcat *	Identical to gunzip -c

Commands above which are appended with '*' come from BusyBox (the Swiss Army Knife of embedded Linux*) http://www.busybox.net/downloads/BusyBox.html*. Others are generic Linux commands and most commands the ***-h*** or ***--help*** argument to provide a terse runtime description of their behavior. More details on the generic Linux commands can found online at *http://en.tldp.org/HOWTO/HOWTO-INDEX/howtos.html* and *http://www.faqs.org/docs/Linux-HOWTO/Remote-Serial-Console-HOWTO.html*

An updated list of the commands may found using ***ls*** command to view all the commands actually available in the */bin* directory in your *console server*.

There were a number of Black Box tools listed above that make it simple to configure the *console server* and make sure the changes are stored in the *console server*'s flash memory, etc. These commands are covered in the previous chapters and include:

- **config** which allows manipulation and querying of the system configuration from the command line. With *config* a new configuration can be activated by running the relevant configurator, which performs the action necessary to make the configuration changes live.
- **portmanager** which provides a buffered interface to each serial port. It is supported by the pmchat and pmshell commands which ensure all serial port access is directed via the *portmanager.*
- **pmpower** is a configurable tool for manipulating remote power devices that are serially or network connected to the *console server*.
- **SDT Connector** is a java client applet that provides point-and-click SSH tunneled connections to the *console server* and Managed Devices.

There are also a number of other CLI commands related to other open source tools embedded in the *console server* including:

- **PowerMan** provides power management for many preconfigured remote power controller (RPC) devices. For CLI details refer *http://linux.die.net/man/1/powerman*
- **Network UPS Tools (NUT)** provides reliable monitoring of UPS and PDU hardware and ensure safe shutdowns of the systems which are connected - with a goal to monitor every kind of UPS and PDU. For CLI details refer *http://www.networkupstools.org*
- **Nagios** is a popular enterprise-class management tool that provides central monitoring of the hosts and services in distributed networks. For CLI details refer http://www.nagios.org

Many components of the *console server* software are licensed under the GNU General Public License (version 2), which Black Box supports. You may obtain a copy of the GNU General Public License at ***http://www.fsf.org/copyleft/gpl.html.*** Black Box will provide source code for any of the components of the software licensed under the GNU General Public License upon request.

The *console server* also embodies the *okvm* console management software. This is GPL code and the full source is available from *http://okvm.sourceforge.net.*

The *console server* BIOS (boot loader code) is a port of *uboot*, which is also a GPL package with source openly available.

The *console server* CGIs (the html code, xml code and web config tools for the Management Console) are proprietary to Black Box, however the code will be provided to customers, under NDA.

Also inbuilt in the *console server* is a Port Manager application and Configuration tools as described in *Chapters 14* and *15*. These both are proprietary to Black Box, but open to customers (as above).

The *console server* also supports GNU *bash* shell script enabling the *Administrator* to run custom scripts. GNU *bash*, version 2.05.0(1)-release (arm-Black Box-linux-gnu) offers the following shell commands:

```
alias [-p] [name[=value] ... ]                  local name[=value] ...
bg [job_spec]                                    logout
bind [-lpvsPVS] [-m keymap] [-f fi break [n]     popd [+N | -N] [-n]
builtin [shell-builtin [arg ...]]                printf format [arguments]
case WORD in [PATTERN [| PATTERN]                pushd [dir | +N | -N] [-n]
cd [-PL] [dir]                                   pwd [-PL]
command [-pVv]                                   read [-ers] [-t timeout] [-p promp]
command [arg ...]                                readonly [-anf] [name ...] or read return [n]
compgen [-abcdefjkvu] [-o option]                select NAME [in WORDS ... ;] do
complete [-abcdefjkvu] [-pr] [-o o]              COMMANDS
continue [n]                                     set [--abefhkmnptuvxBCHP] [-o opti]
declare [-afFrxi] [-p] name[=value]              shift [n]
dirs [-clpv] [+N] [-N]                           shopt [-pqsu] [-o long-option] opt
disown [-h] [-ar] [jobspec ...]                  source filename
echo [-neE] [arg ...]                            suspend [-f]
enable [-pnds] [-a] [-f filename]                test [expr]
eval [arg ...]                                   time [-p] PIPELINE
exec [-cl] [-a name] file [redirec]              times
exit [n]                                         trap [arg] [signal_spec ...]
export [-nf] [name ...] or export                true
```

false
fc [-e ename] [-nlr] [first] [last]
fg [job_spec]
for NAME [in WORDS ... ;] do COMMA
function NAME { COMMANDS ; } or NA
getopts optstring name [arg]
hash [-r] [-p pathname] [name ...]
help [-s] [pattern ...]
history [-c] [-d offset] [n] or hi
if COMMANDS; then COMMANDS; [elif
jobs [-lnprs] [jobspec ...] or *job kill [-s sigspec | -n signum | -si let arg [arg ...]*
type [-apt] name [name ...]
typeset [-afFrxi] [-p] name[=value ulimit [-SHacdflmnpstuv] [limit]
umask [-p] [-S] [mode]
unalias [-a] [name ...]
unset [-f] [-v] [name ...]
until COMMANDS; do COMMANDS; done
variables - Some variable names an wait [n]
while COMMANDS; do COMMANDS; done { COMMANDS ; }

Appendix B Hardware Specifications

FEATURE	VALUE
Dimensions	LES1208A/16A/48A: 17 x 12 x 1.75 in (43.2 x 31.3. x 4.5 cm) LES1116A/48A: 17 x 8.5 x 1.75 in (43.2 x 21. x 4.5 cm) LES1108A: 8.2 x 4.9 x 1.2 in (20.8 x 12.6 x 4.5 cm)
Weight	LES1208A/16A/48A: 5.4 kg (11.8 lbs) LES1116A/48A: 3.9 kg (8.5 lbs) LES1108A: 1.7 kg (3.7 lbs)
Ambient operating temperature	5°C to 50°C (41°F to 122°F)
Non operating storage temperature	-30°C to +60°C (-20°F to +140°F)
Humidity	5% to 90%
Power	Refer to Chapter 2 for various models
Power Consumption	All less than 30W
CPU	Micrel KS8695P controller
Memory	LES1208A/16A/48A: 64MB SDRAM 16MB Flash 512MB USB Flash LES1116A/48A: 64MB SDRAM 16MB Flash LES1108A:: 16MB SDRAM 8MB Flash
Serial Connectors	LES1208A: 8 RJ-45 RS-232 serial ports LES1216A: 16 RJ-45 RS-232 serial ports LES1248A: 48 RJ-45 RS-232 serial ports LES1116A: 16 RJ-45 RS-232 serial ports LES1148A: 48 RJ-45 RS-232 serial ports LES1108A 8 RJ-45 RS-232 serial ports All models: 1 DB-9 RS-232 console/ modem serial port
Serial Baud Rates	RJ45 ports - 50 to 230,400bps) DB9 port - 2400 to 115,200 bps
Ethernet Connectors	LES1208A/16A/48A: Two RJ-45 10/100Base-T Ethernet ports LES1108A/16A/48A: One RJ-45 10/100Base-T Ethernet ports

Appendix C Safety & Certifications

Please take care to follow the safety precautions below when installing and operating the *console server*:

- Do not remove the metal covers. There are no operator serviceable components inside. Opening or removing the cover may expose you to dangerous voltage which may cause fire or electric shock. Refer all service to Black Box qualified personnel.

- To avoid electric shock the power cord protective grounding conductor must be connected through to ground.

- Always pull on the plug, not the cable, when disconnecting the power cord from the socket.

Do not connect or disconnect the *console server* during an electrical storm. We recommend that you use a surge suppressor or UPS to protect the equipment from transients.

FCC Warning Statement

This device complies with Part 15 of the FCC rules. Operation of this device is subject to the following conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference that may cause undesired operation.

Appendix F End User License Agreement

READ BEFORE USING THE ACCOMPANYING SOFTWARE

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE USING THE ACCOMPANYING SOFTWARE, THE USE OF WHICH IS LICENSED FOR USE ONLY AS SET FORTH BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE. IF YOU USE ANY PART OF THE SOFTWARE, SUCH USE WILL INDICATE THAT YOU ACCEPT THESE TERMS.

You have acquired a product that includes Black Box ("Black Box") proprietary software and/or proprietary software licensed to Black Box. This Black Box End User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Black Box for the installed software product of Black Box origin, as well as associated media, printed materials, and "online" or electronic documentation ("Software"). By installing, copying, downloading, accessing, or otherwise using the Software, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, Black Box is not willing to license the Software to you. In such event, do not use or install the Software. If you have purchased the Software, promptly return the Software and all accompanying materials with proof of purchase for a refund.

Products with separate end user license agreements that may be provided along with the Software are licensed to you under the terms of those separate end user license agreements.

LICENSE GRANT. Subject to the terms and conditions of this EULA, Black Box grants you a nonexclusive right and license to install and use the Software on a single CPU, provided that, (1) you may not rent, lease, sell, sublicense or lend the Software; (2) you may not reverse engineer, decompile, disassemble or modify the Software, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation; and (3) you may not transfer rights under this EULA unless such transfer is part of a permanent sale or transfer of the Product, you transfer at the same time all copies of the Software to the same party or destroy such materials not transferred, and the recipient agrees to this EULA.

No license is granted in any of the Software's proprietary source code. This license does not grant you any rights to patents, copyright, trade secrets, trademarks or any other rights with respect to the Software.

You may make a reasonable number of copies of the electronic documentation accompanying the Software for each Software license you acquire, provided that, you must reproduce and include all copyright notices and any other proprietary rights notices appearing on the electronic documentation. Black Box reserves all rights not expressly granted herein.

INTELLECTUAL PROPERTY RIGHTS. The Software is protected by copyright laws, international copyright treaties, and other intellectual property laws and treaties. Black Box and its suppliers retain all ownership of, and intellectual property rights in (including copyright), the Software components and all copies thereof, provided however, that (1) certain components of the Software, including *SDT Connector*, are components licensed under the GNU General Public License Version 2, which Black Box supports, and (2) the *SDT Connector* includes code from JSch, a pure Java implementation of SSH2 which is licensed under BSD style license. Copies of these licenses are detailed below and Black Box will provide source code for any of the components of the Software licensed under the GNU General Public License upon request.

EXPORT RESTRICTIONS. You agree that you will not export or re-export the Software, any part thereof, or any process or service that is the direct product of the Software in violation of any applicable laws or regulations of the United States or the country in which you obtained them.

U.S. GOVERNMENT RESTRICTED RIGHTS. The Software and related documentation are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software – Restricted Rights at 48 C.F.R. 52.227-19, as applicable, or any successor regulations.

TERM AND TERMINATION. This EULA is effective until terminated. The EULA terminates immediately if you fail to comply with any term or condition. In such an event, you must destroy all copies of the Software. You may also terminate this EULA at any time by destroying the Software.

GOVERNING LAW AND ATTORNEY'S FEES. This EULA is governed by the laws of the State of Utah, USA, excluding its conflict of law rules. You agree that the United Nations Convention on Contracts for the International

Sale of Goods is hereby excluded in its entirety and does not apply to this EULA. If you acquired this Software in a country outside of the United States, that country's laws may apply. In any action or suit to enforce any right or remedy under this EULA or to interpret any provision of this EULA, the prevailing party will be entitled to recover its costs, including reasonable attorneys' fees.

ENTIRE AGREEMENT. This EULA constitutes the entire agreement between you and Black Box with respect to the Software, and supersedes all other agreements or representations, whether written or oral. The terms of this EULA can only be modified by express written consent of both parties. If any part of this EULA is held to be unenforceable as written, it will be enforced to the maximum extent allowed by applicable law, and will not affect the enforceability of any other part.

Should you have any questions concerning this EULA, or if you desire to contact Black Box for any reason, please contact the Black Box representative serving your company.

THE FOLLOWING DISCLAIMER OF WARRANTY AND LIMITATION OF LIABILITY IS INCORPORATED INTO THIS EULA BY REFERENCE. THE SOFTWARE IS NOT FAULT TOLERANT. YOU HAVE INDEPENDENTLY DETERMINED HOW TO USE THE SOFTWARE IN THE DEVICE, AND BLACK BOX HAS RELIED UPON YOU TO CONDUCT SUFFICIENT TESTING TO DETERMINE THAT THE SOFTWARE IS SUITABLE FOR SUCH USE.

LIMITED WARRANTY Black Box warrants the media containing the Software for a period of ninety (90) days from the date of original purchase from Black Box or its authorized retailer. Proof of date of purchase will be required. Any updates to the Software provided by Black Box (which may be provided by Black Box at its sole discretion) shall be governed by the terms of this EULA. In the event the product fails to perform as warranted, Black Box's sole obligation shall be, at Black Box's discretion, to refund the purchase price paid by you for the Software on the defective media, or to replace the Software on new media. Black Box makes no warranty or representation that its Software will meet your requirements, will work in combination with any hardware or application software products provided by third parties, that the operation of the software products will be uninterrupted or error free, or that all defects in the Software will be corrected.

BLACK BOX DISCLAIMS ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OTHER THAN AS STATED HEREIN, THE ENTIRE RISK AS TO SATISFACTORY QUALITY, PERFORMANCE, ACCURACY, AND EFFORT IS WITH YOU. ALSO, THERE IS NO WARRANTY AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE SOFTWARE OR AGAINST INFRINGEMENT. IF YOU HAVE RECEIVED ANY WARRANTIES REGARDING THE DEVICE OR THE SOFTWARE, THOSE WARRANTIES DO NOT ORIGINATE FROM, AND ARE NOT BINDING ON, BLACK BOX.

NO LIABILITY FOR CERTAIN DAMAGES. EXCEPT AS PROHIBITED BY LAW, BLACK BOX SHALL HAVE NO LIABILITY FOR COSTS, LOSS, DAMAGES OR LOST OPPORTUNITY OF ANY TYPE WHATSOEVER, INCLUDING BUT NOT LIMITED TO, LOST OR ANTICIPATED PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, EXEMPLARY SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY OR OTHERWISE ARISING FROM OR IN CONNECTION WITH THIS EULA OR THE USE OR PERFORMANCE OF THE SOFTWARE. IN NO EVENT SHALL BLACK BOX BE LIABLE FOR ANY AMOUNT IN EXCESS OF THE LICENSE FEE PAID TO BLACK BOX UNDER THIS EULA. SOME STATES AND COUNTRIES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION MAY NOT APPLY TO YOU.

JSch License

SDT Connector includes code from JSch, a pure Java implementation of SSH2. JSch is licensed under BSD style license and it is:

Copyright (c) 2002, 2003, 2004 Atsuhiko Yamanaka, JCraft,Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SDT Connector License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

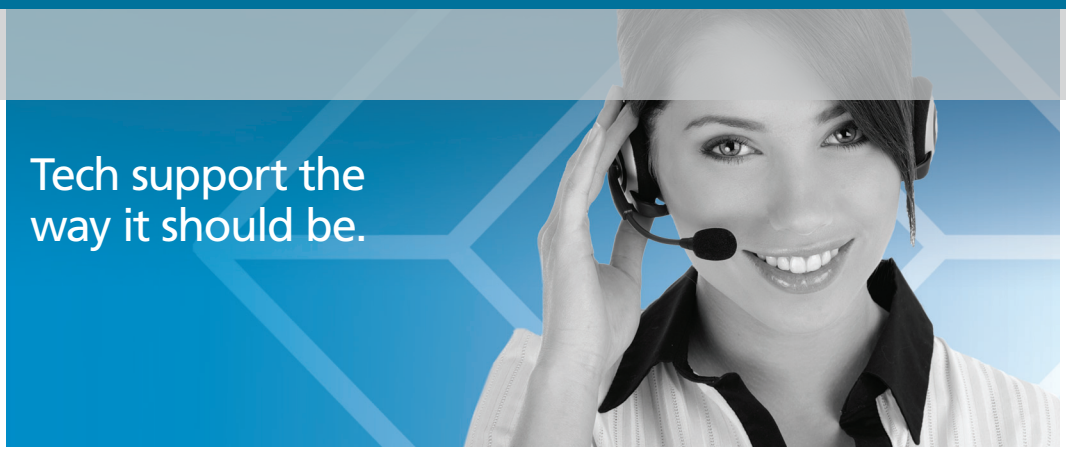
11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Black Box Tech Support: FREE! Live. 24/7.



Great tech support is just 20 seconds away at 724-746-5500 or blackbox.com.

BLACK BOX®
NETWORK SERVICES

About Black Box

Black Box Network Services is your source for more than 118,000 networking and infrastructure products. You'll find everything from cabinets and racks and power and surge protection products to media converters and Ethernet switches all supported by free, live 24/7 Tech support available in 20 seconds or less.

© Copyright 2009. All rights reserved. Black Box® and the Double Diamond logo are registered trademarks of BB Technologies, Inc. Any third-party trademarks appearing in this white paper are acknowledged to be the property of their respective owners.